

# Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP04/019405

International filing date: 24 December 2004 (24.12.2004)

Document type: Certified copy of priority document

Document details: Country/Office: JP  
Number: 2004-114637  
Filing date: 08 April 2004 (08.04.2004)

Date of receipt at the International Bureau: 10 March 2005 (10.03.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland  
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

28.12.2004

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日            2 0 0 4 年   4 月   8 日  
Date of Application:

出 願 番 号            特 願 2 0 0 4 - 1 1 4 6 3 7  
Application Number:  
[ST. 10/C] :            [ J P 2 0 0 4 - 1 1 4 6 3 7 ]

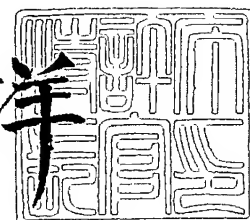
出      願      人            新日本製鐵株式会社  
Applicant(s):



2 0 0 5 年   2 月 2 4 日

特許庁長官  
Commissioner,  
Japan Patent Office

小 川 洋



【書類名】 特許願  
【整理番号】 NA400566  
【提出日】 平成16年 4月 8日  
【あて先】 特許庁長官殿  
【国際特許分類】 G06F 17/60  
【発明者】  
    【住所又は居所】 富津市新富 2 0 - 1 新日本製鐵株式会社 技術開発本部内  
    【氏名】 屋地 靖人  
【発明者】  
    【住所又は居所】 富津市新富 2 0 - 1 新日本製鐵株式会社 技術開発本部内  
    【氏名】 杉山 賢司  
【発明者】  
    【住所又は居所】 富津市新富 2 0 - 1 新日本製鐵株式会社 技術開発本部内  
    【氏名】 黒川 哲明  
【発明者】  
    【住所又は居所】 富津市新富 2 0 - 1 新日本製鐵株式会社 技術開発本部内  
    【氏名】 塩谷 政典  
【発明者】  
    【住所又は居所】 富津市新富 2 0 - 1 新日本製鐵株式会社 技術開発本部内  
    【氏名】 伊藤 邦春  
【発明者】  
    【住所又は居所】 富津市新富 2 0 - 1 新日本製鐵株式会社 技術開発本部内  
    【氏名】 小林 敬和  
【特許出願人】  
    【識別番号】 000006655  
    【氏名又は名称】 新日本製鐵株式会社  
【代理人】  
    【識別番号】 100090273  
    【弁理士】  
    【氏名又は名称】 國分 孝悦  
    【電話番号】 03-3590-8901  
【手数料の表示】  
    【予納台帳番号】 035493  
    【納付金額】 16,000円  
【提出物件の目録】  
    【物件名】 特許請求の範囲 1  
    【物件名】 明細書 1  
    【物件名】 図面 1  
    【物件名】 要約書 1  
    【包括委任状番号】 9707819

**【書類名】 特許請求の範囲****【請求項 1】**

生産・物流プロセスの物流状態と物流制約を表現した生産・物流プロセスを模擬する離散系の生産・物流シミュレータと、

上記生産・物流プロセスの物流状態と物流制約とを数式で表現した数式モデルであって、着目している物流のスケジュールを作成するのに係わる情報を取り込んで作成された状態方程式を用いた数式モデルを保持する数式モデル保持装置と、

上記数式モデルに対して所定の評価関数を用いて最適化計算処理を行って上記生産・物流シミュレータに対する物流指示を算出する最適化計算装置とを有し、

上記最適化計算処理によりフィードバックゲインを算出し、そのフィードバックゲインと物流状態とを用いて物流指示を算出して上記生産・物流シミュレータに与えて、シミュレーションを進め、新たな物流状態を得て、この新たな物流状態をもとに新たな物流指示を算出するという処理を繰り返すことにより得られたシミュレーション結果から上記生産・物流プロセスにおける生産・物流スケジュールを作成するようにしたことを特徴とする生産・物流スケジュール作成装置。

**【請求項 2】**

上記離散系の生産・物流シミュレータはグラフモデルを用いてシミュレータを構築するものであり、上記状態方程式はグラフモデルの状態方程式であることを特徴とする請求項 1 に記載の生産・物流スケジュール作成装置。

**【請求項 3】**

上記離散系の生産・物流シミュレータはグラフモデルを用いてシミュレータを構築するものであり、上記状態方程式は処理時間に比例したきざみ時間遅れに従って表わされる接続行列と遷移行列とを用いた状態方程式であることを特徴とする請求項 1 に記載の生産・物流スケジュール作成装置。

**【請求項 4】**

上記最適化計算処理によりフィードバックゲインを算出し、そのフィードバックゲインと物流状態とを用いて物流指示を算出する場合に、フィードバックゲインと工程内製品仕掛状態を表わす状態ベクトルとから、製品の移動操作端に対する操作ベクトルを求めたのち、各移動操作端に対して、上記求めた操作ベクトルの中で正值の大きい操作量が得られら製品順に各製品を移動可能な数だけ移動させるように操作することを特徴とする請求項 2 又は 3 に記載の生産・物流スケジュール作成装置。

**【請求項 5】**

上記離散系の生産・物流シミュレータはペトリネットモデルを用いてシミュレータを構築するものであり、上記状態方程式はペトリネットモデルの状態方程式であることを特徴とする請求項 1 に記載の生産・物流スケジュール作成装置。

**【請求項 6】**

上記離散系の生産・物流シミュレータはペトリネットモデルを用いてシミュレータを構築するものであり、上記状態方程式は処理時間に比例したきざみ時間遅れに従って表わされる接続行列と遷移行列とを用いた状態方程式であることを特徴とする請求項 1 に記載の生産・物流スケジュール作成装置。

**【請求項 7】**

上記最適化計算処理によりフィードバックゲインを算出し、そのフィードバックゲインと物流状態とを用いて物流指示を算出する場合に、フィードバックゲインと工程内製品仕掛状態を表わす状態ベクトルとから、製品の移動操作端に対する操作ベクトルを求めたのち、各移動操作端に対して、上記求めた操作ベクトルの中で正值の大きい操作量が得られら製品順に各製品を移動可能な数だけ移動させるように操作することを特徴とする請求項 5 又は 6 に記載の生産・物流スケジュール作成装置。

**【請求項 8】**

上記最適化計算処理は線形二次制御を利用するものであり、所定の評価行列を用いて表わされる評価関数を最小とするように前記フィードバックゲインを算出することを特徴と

する請求項 1 ～ 7 のいずれか 1 項に記載の生産・物流スケジュール作成装置。

【請求項 9】

生産・物流プロセスの物流状態と物流制約を表現した生産・物流プロセスを模擬する離散系の生産・物流シミュレータと、

上記生産・物流プロセスの物流状態と物流制約とを数式で表現した数式モデルであって、着目している物流のスケジュールを作成するのに係わる情報を取り込んで作成された状態方程式を用いた数式モデルを保持する数式モデル保持装置と、

上記数式モデルに対して所定の評価関数を用いて最適化計算処理を行って上記生産・物流シミュレータに対する物流指示を算出する最適化計算装置とを有し、

上記最適化計算処理によりフィードバックゲインを算出し、そのフィードバックゲインと物流状態とを用いて物流指示を算出して上記生産・物流シミュレータに与えて、シミュレーションを進め、新たな物流状態を得て、この新たな物流状態をもとに新たな物流指示を算出するという処理を繰り返すことにより得られたシミュレーション結果に基づいて生産・物流プロセスの制御を行うことを特徴とする生産・物流プロセス制御装置。

【請求項 1 0】

生産・物流プロセスの物流状態と物流制約を表現した生産・物流プロセスを模擬する離散系の生産・物流シミュレータと、上記生産・物流プロセスの物流状態と物流制約とを数式で表現した数式モデルであって、着目している物流のスケジュールを作成するのに係わる情報を取り込んで作成された状態方程式を用いた数式モデルを保持する数式モデル保持装置と、上記数式モデルに対して所定の評価関数を用いて最適化計算処理を行って上記生産・物流シミュレータに対する物流指示を算出する最適化計算装置とを有する生産・物流スケジュール作成装置により生産・物流スケジュールを作成する方法であって、

上記最適化計算処理によりフィードバックゲインを算出し、そのフィードバックゲインと物流状態とを用いて物流指示を算出して上記生産・物流シミュレータに与えて、シミュレーションを進め、新たな物流状態を得て、この新たな物流状態をもとに新たな物流指示を算出するという処理を繰り返すことにより得られたシミュレーション結果から上記生産・物流プロセスにおける生産・物流スケジュールを作成するようにしたことを特徴とする生産・物流スケジュール作成方法。

【請求項 1 1】

生産・物流プロセスの物流状態と物流制約を表現した生産・物流プロセスを模擬する離散系の生産・物流シミュレータと、上記生産・物流プロセスの物流状態と物流制約とを数式で表現した数式モデルであって、着目している物流のスケジュールを作成するのに係わる情報を取り込んで作成された状態方程式を用いた数式モデルを保持する数式モデル保持装置と、上記数式モデルに対して所定の評価関数を用いて最適化計算処理を行って上記生産・物流シミュレータに対する物流指示を算出する最適化計算装置とを有する生産・物流プロセス制御装置により生産・物流プロセスを制御する方法であって、

上記最適化計算処理によりフィードバックゲインを算出し、そのフィードバックゲインと物流状態とを用いて物流指示を算出して上記生産・物流シミュレータに与えて、シミュレーションを進め、新たな物流状態を得て、この新たな物流状態をもとに新たな物流指示を算出するという処理を繰り返すことにより得られたシミュレーション結果に基づいて生産・物流プロセスの制御を行うようにしたことを特徴とする生産・物流プロセス制御方法。

【請求項 1 2】

請求項 1 0 に記載の生産・物流スケジュール作成方法をコンピュータに実行させることを特徴とするコンピュータプログラム。

【請求項 1 3】

請求項 1 1 に記載の生産・物流プロセス制御方法をコンピュータに実行させることを特徴とするコンピュータプログラム。

【請求項 1 4】

請求項 1 2 又は 1 3 に記載のコンピュータプログラムを記録したことを特徴とするコン

コンピュータ読み取り可能な記録媒体。

**【書類名】 明細書**

**【発明の名称】** 生産・物流スケジュール作成装置及び方法、生産・物流プロセス制御装置及び方法、コンピュータプログラム、及びコンピュータ読み取り可能な記録媒体

**【技術分野】****【0001】**

本発明は生産・物流スケジュール作成装置及び方法、生産・物流プロセス制御装置及び方法、コンピュータプログラム、及びコンピュータ読み取り可能な記録媒体に関し、特に、操作者の熟練度に依存することなく対象システムのスケジュールを正確に作成するような場合に用いて好適なものである。

**【背景技術】****【0002】**

従来、見込み生産を行わない受注生産の形態では、受注した品目を製造オーダーに分割し、納期遵守率や設備稼働率、中間在庫量、コスト等の観点 considering して生産・物流スケジュールを作成している。上記生産・物流スケジュールを作成する手法として、技術的に大別すると2つの手法が用いられている。

**【0003】**

すなわち、第1の手法としては、例えば、特許文献1の「生産計画評価方法及びシステム」に開示されているように、コンピュータ上に構築した工場を模したシミュレーション上で、実機器と同じインタフェースから取得した情報を使用して実機器の稼働を予測し、稼働予測に基づいて、実機器より速い速度で仮想的な生産を行い、仮想的な生産の過程及び結果を用いて、精度の高い指標を提示することによって、生産計画の評価及び選択を可能にする手法である。

**【0004】**

また、第2の手法としては、特許文献2の「物流計画作成装置」にて開示されているように、線形計画法、数理計画法等のように、最適性が保証される手法に基づいてスケジュールを作成する手法である。

**【0005】**

**【特許文献1】** 特開2002-366219号公報

**【特許文献2】** 特開2000-172745号公報

**【発明の開示】****【発明が解決しようとする課題】****【0006】**

上記特許文献1に記載の「生産計画評価方法及びシステム」に開示されているように、シミュレータを用いて生産・物流スケジュールを作成する手法は、満足できる結果が得られるまでには、(1) 条件を種々に変えながらシミュレーションを行い、その結果の評価を何回も繰返し行う必要があった。したがって、(2) 大規模工場では生産・物流スケジュールを作成するのに多くの時間がかかってしまう問題点があった。また、(3) 高精度な生産・物流スケジュールを得るためには、シミュレーション・ルールを細かく設定しなければならない問題点があった。

**【0007】**

また、上記特許文献2の「物流計画作成装置」にて開示されているように、線形計画法、数理計画法等のように、最適性が保証される手法に基づいてスケジュールを作成する手法の場合には、(1) 生産・物流スケジュールを作成する規模が大きくなると、実用的な時間内に解くことが困難になってしまう問題点があった。また、(2) 数式で記述できない制約や条件に起因する誤差が生じるため、得られた生産・物流スケジュールが実行可能であるかどうかは保証されていなかった。

**【0008】**

本発明は上述の問題点に鑑みてなされたものであり、シミュレーションを繰返し行うことなく最適な生産・物流シミュレーション結果を得ることができるようにして、スケジュール作成対象の生産・物流プロセスで実際に使用可能であることが保証された生産・物

流スケジュールを高速に、且つ高精度に作成できるようにすることを目的とする。

【課題を解決するための手段】

【0009】

本発明の生産・物流スケジュール作成装置は、生産・物流プロセスの物流状態と物流制約を表現した生産・物流プロセスを模擬する離散系の生産・物流シミュレータと、上記生産・物流プロセスの物流状態と物流制約とを数式で表現した数式モデルであって、着目している物流のスケジュールを作成するのに係わる情報を取り込んで作成された状態方程式を用いた数式モデルを保持する数式モデル保持装置と、上記数式モデルに対して所定の評価関数を用いて最適化計算処理を行って上記生産・物流シミュレータに対する物流指示を算出する最適化計算装置とを有し、上記最適化計算処理によりフィードバックゲインを算出し、そのフィードバックゲインと物流状態とを用いて物流指示を算出して上記生産・物流シミュレータに与えて、シミュレーションを進め、新たな物流状態を得て、この新たな物流状態をもとに新たな物流指示を算出するという処理を繰り返すことにより得られたシミュレーション結果から上記生産・物流プロセスにおける生産・物流スケジュールを作成するようにした点に特徴を有する。

本発明の生産・物流プロセス制御装置は、生産・物流プロセスの物流状態と物流制約を表現した生産・物流プロセスを模擬する離散系の生産・物流シミュレータと、上記生産・物流プロセスの物流状態と物流制約とを数式で表現した数式モデルであって、着目している物流のスケジュールを作成するのに係わる情報を取り込んで作成された状態方程式を用いた数式モデルを保持する数式モデル保持装置と、上記数式モデルに対して所定の評価関数を用いて最適化計算処理を行って上記生産・物流シミュレータに対する物流指示を算出する最適化計算装置とを有し、上記最適化計算処理によりフィードバックゲインを算出し、そのフィードバックゲインと物流状態とを用いて物流指示を算出して上記生産・物流シミュレータに与えて、シミュレーションを進め、新たな物流状態を得て、この新たな物流状態をもとに新たな物流指示を算出するという処理を繰り返すことにより得られたシミュレーション結果に基づいて生産・物流プロセスの制御を行う点に特徴を有する。

本発明の生産・物流スケジュール作成方法は、生産・物流プロセスの物流状態と物流制約を表現した生産・物流プロセスを模擬する離散系の生産・物流シミュレータと、上記生産・物流プロセスの物流状態と物流制約とを数式で表現した数式モデルであって、着目している物流のスケジュールを作成するのに係わる情報を取り込んで作成された状態方程式を用いた数式モデルを保持する数式モデル保持装置と、上記数式モデルに対して所定の評価関数を用いて最適化計算処理を行って上記生産・物流シミュレータに対する物流指示を算出する最適化計算装置とを有する生産・物流スケジュール作成装置により生産・物流スケジュールを作成する方法であって、上記最適化計算処理によりフィードバックゲインを算出し、そのフィードバックゲインと物流状態とを用いて物流指示を算出して上記生産・物流シミュレータに与えて、シミュレーションを進め、新たな物流状態を得て、この新たな物流状態をもとに新たな物流指示を算出するという処理を繰り返すことにより得られたシミュレーション結果から上記生産・物流プロセスにおける生産・物流スケジュールを作成するようにした点に特徴を有する。

本発明の生産・物流プロセス制御方法は、生産・物流プロセスの物流状態と物流制約を表現した生産・物流プロセスを模擬する離散系の生産・物流シミュレータと、上記生産・物流プロセスの物流状態と物流制約とを数式で表現した数式モデルであって、着目している物流のスケジュールを作成するのに係わる情報を取り込んで作成された状態方程式を用いた数式モデルを保持する数式モデル保持装置と、上記数式モデルに対して所定の評価関数を用いて最適化計算処理を行って上記生産・物流シミュレータに対する物流指示を算出する最適化計算装置とを有する生産・物流プロセス制御装置により生産・物流プロセスを制御する方法であって、上記最適化計算処理によりフィードバックゲインを算出し、そのフィードバックゲインと物流状態とを用いて物流指示を算出して上記生産・物流シミュレータに与えて、シミュレーションを進め、新たな物流状態を得て、この新たな物流状態をもとに新たな物流指示を算出するという処理を繰り返すことにより得られたシミュレーション



ョン結果に基づいて生産・物流プロセスの制御を行うようにした点に特徴を有する。

本発明のコンピュータプログラムは、上記本発明の生産・物流スケジュール作成方法又は上記本発明の生産・物流プロセス制御方法をコンピュータに実行させる点に特徴を有する。

本発明のコンピュータ読み取り可能な記録媒体は、上記本発明のコンピュータプログラムを記録した点に特徴を有する。

#### 【発明の効果】

##### 【0010】

本発明によれば、詳細シミュレーションを実行中に物流指示が必要な事象が発生するたびにシミュレータからシミュレーション現時点での物流状態及び物流制約の情報を検出し、上記検出した検出情報と予め定めた評価指標を元に最適化手法によって最適物流指示を計算し、上記計算結果に基づいて以後の詳細シミュレーションを進め、次に物流指示が必要な事象が発生した時点で最適物流指示を再度計算することを繰り返し行うようにしたので、詳細シミュレーション自体は一度行うだけで最適物流指示を得ることができる。また、シミュレータと、数式モデルと、最適化装置とを連動させて最適物流指示を計算し、上記計算結果のシミュレーションを行ってスケジュールを作成するようにしたので、物流制約条件が複雑であっても実行可能なスケジュールを作成することができる。また、所望の評価指標を最良にするスケジュールを作成できるとともに、計算時間を短縮して実用的な時間内にスケジュールを作成することができる。これらにより、最適性と実行可能性の両方を確保できる。また、対象とする生産・物流プロセスの状態に応じたスケジュールを作成する際の手間及び時間を大幅に削減することができる。さらに、スケジュールを作成する対象の規模が大きい場合においても、正確な生産・物流スケジュールを高速に、且つ高精度に作成することが可能となる。

#### 【発明を実施するための最良の形態】

##### 【0011】

以下、図面を参照して、本発明の好適な実施形態について説明する。

##### (第1の実施形態)

図1は、本発明の第1の実施形態を示し、物流スケジュール作成装置の一例を説明するブロック図である。図1に示したように、本実施形態の物流スケジュール作成装置は、物流シミュレータ1aを備えたシミュレーション制御部1、最適化計算装置2等によって構成されている。

##### 【0012】

本実施形態では、物流シミュレータ1aは生産・物流プロセスの物流状態と物流制約を表現した生産・物流プロセスを模擬したシミュレータ、いわば工場を模擬した大型のシミュレータであり、本実施形態においては、グラフモデルを用いてシミュレータを構築するもので、事象（シミュレータのイベント）毎に物を動かす離散系として構成されている。

##### 【0013】

また、上記物流シミュレータ1aに対応させて数学モデル（数式モデル）3が構成されている。本実施形態においては、生産・物流プロセスの物流状態及び物流制約の中から、作成する物流スケジュールに関連する要素を取り込んで、下式(イ)に示すような状態方程式を用いて上記数式モデル3が作成されている。上記数式モデル3は、半導体記憶手段等により構成される数式モデル保持手段（図示せず）によって保持されている。

$$M(k+1) = M(k) + b \cdot u(k) \cdots (イ)$$

すなわち、離散化したある時刻kにおいて各点に存在する要素の数を示す状態ベクトル、言い換えれば状態量をM(k)で表し、各線の発火の有無を"1"及び"0"で表現した操作ベクトルをu(k)で表わすと、次の時刻k+1における状態ベクトルM(k+1)は、接続行列bを用いて上式(イ)で表わすことができる。

##### 【0014】

最適化計算装置2では、上記数式モデル3に対して最適化計算処理を行い、フィードバックゲインKを算出するようにしている。上記最適化計算装置2によって行われる最適化

計算は線形二次 (LQ) 制御を利用するものであり、下式(ロ)に示すような評価関数 S を用いて行われる。

$$S = \sum \{M'QM + u'Ru\} \cdots (\text{ロ})$$

評価関数 S において、Q, R は制御目的にあわせて設定された適当な行列であり、M', u' は、それぞれ状態ベクトル M、操作ベクトル u の転置ベクトルである。そして、評価関数 S が最小となるように制御することを考えれば、

$$u(k) = -K \cdot M(k) \cdots (\text{ハ})$$

とした状態フィードバック制御を行うフィードバックゲイン K を最適制御理論より求めることができる。

#### 【0015】

シミュレーション制御部 1 の最適制御方策部 1b では、最適化計算装置 2 で算出されるフィードバックゲイン K と物流状態 (状態ベクトル M) とを用いて物流指示 (操作ベクトル u) を算出して物流シミュレータ 1a に与えて、シミュレーションを進め、新たな物流状態を得て、この新たな物流状態をもとに新たな物流指示を算出するという処理を繰り返す。そして、これにより得られたシミュレーション結果 4 から生産・物流プロセスにおける物流スケジュールを作成する。

#### 【0016】

以下に、第 1 の実施形態の具体例を、図 2～5 を参照しながら説明する。図 2 に示したように、第 1 に、生産・物流プロセスのグラフモデルとして、処理時間  $T_p$  に比例した数の点で各工程を表わしたグラフモデルを製品種ごとに構築する (ステップ S1)。

#### 【0017】

第 2 に、構築したグラフモデルの状態方程式と、設定した評価関数 Q, R とからフィードバックゲイン行列 K を製品種ごとに求める (ステップ S2)。

#### 【0018】

第 3 に、この求めたフィードバックゲイン行列 K と工程内製品仕掛状態を表す状態ベクトル  $M(k)$  とから、製品の移動操作端に対する操作ベクトル  $u(k)$  を製品種ごとに求めたのち、仮想的な生産・物流プロセス内の各移動操作端に対して、正值で大きい操作量が得られた製品順に移動可能な数だけ移動操作を順次行うという方法を用いて、所定期間の生産・物流プロセスのシミュレーションを実行する (ステップ S3)。

#### 【0019】

図 3 は、図 2 に示したステップ S1 の処理、すなわち、生産・物流プロセスのグラフモデルとして、処理時間を比例した数の点で各工程を表したグラフモデルを製品種ごとに構築するための処理手順の例を示すフローチャートである。

#### 【0020】

すなわち、まず離散きざみ時間  $\Delta T$  を入力し (ステップ S101)、次に製品種ごとに全ての処理工程と処理時間  $T_p$  とを入力する (ステップ S102)。そして、処理工程ごとに処理時間  $T_p$  をきざみ時間  $\Delta T$  で除して整数化することにより上記各処理工程の点の数  $n_p$  を求め (ステップ S103、S104)、全処理工程の点を線で連結して製品種ごとのグラフモデルを構築する (ステップ S105)。

#### 【0021】

そして、全ての製品についてグラフモデルを構築するまでステップ S106 からステップ S102 の処理に戻り、以上の処理を繰り返して行う。なお、離散きざみ時間  $\Delta T$  は、各製品種各工程処理時間の最大公約数をもって定義するのが最も効率的であるが、目的とする制御精度を勘案して適宜設定すればよい。

#### 【0022】

図 4 は、図 2 に示したステップ S2 の処理、すなわち、構築したグラフモデルの状態方程式と設定した評価関数とからフィードバックゲイン行列 K を製品種ごとに求めるための処理手順の例を示すフローチャートである。なお、全製品についてグラフモデルの接続行列 b 作成し、それを最適化計算装置 2 に入力し、全製品を一括で計算するようにしてもよい。

**【0023】**

すなわち、まず製品種ごとにグラフモデルの状態方程式を記述する接続行列  $b$  を最適化計算装置 2 に入力するとともに（ステップ S 201）、評価関数を表す行列  $Q$ ,  $R$  を入力する（ステップ S 202）。そして、上記入力した接続行列  $b$  及び評価行列  $Q$ ,  $R$  からフィードバックゲイン行列  $K$  を計算する（ステップ S 203）。この計算を全ての製品について完了するまでステップ S 204 からステップ S 201 の処理に戻り、以上の処理を繰り返して行う。

**【0024】**

図 5 は、図 2 に示したステップ S 3 の処理、すなわち、上述のようにして計算したフィードバックゲイン行列  $K$  と工程内製品仕掛状態を表す状態ベクトル  $M(K)$  とから、製品の移動操作端に対する操作ベクトル  $u(K)$  を製品種ごとに求めたのち、仮想的な生産・物流プロセス内の各移動操作端に対して、正值で大きい操作量が得られた製品順に移動可能な数だけ移動操作を順次行うという方法を用いて、所定期間の生産・物流プロセスのシミュレーションを実行するための処理手順の例を示すフローチャートである。

**【0025】**

すなわち、まずグラフモデルの初期状態における状態ベクトル  $M(0)$  を入力するとともに、投入予定の全製品（要素）を投入点に入力し、時刻  $k$  の値を 0 に初期化したのち（ステップ S 301）、現時点（時刻： $k$ ）における全製品種の状態ベクトル  $M(k)$  を入力する（ステップ S 302）。

**【0026】**

そして、上述の方法を用いて計算されたフィードバックゲイン行列  $K$  と上記入力した状態ベクトル  $M(k)$  とをかけて製品種ごとに操作ベクトル  $u(k)$  を計算し（ステップ S 303）、次に移動操作端ごとに、正值で大きい操作量が得られた製品順に、移動可能な数だけ移動操作を実行する（ステップ S 304）。

**【0027】**

次に、時刻を進めたのち（ステップ S 305）、各ステップのシミュレーションが終了したかどうかを判断し（ステップ S 306）、終了していないときはステップ S 302 の処理に戻る。一方、終了したときは、そのシミュレーション結果をもって生産スケジュールとする（ステップ S 307）。

**【0028】**

上述したように、本実施形態においては、最適化計算を行った結果得られた指示と、その事象における状態とに基づいた最適制御を行うので、それぞれの事象毎に最適化計算を実行したシミュレーション結果に基づいてスケジュールを作成することができ、しかも、そのスケジュールは生産・物流プロセスの制約を表現したシミュレータによるシミュレーション結果であるから、実際に使用可能であることを確認することができる。

**【0029】**

これにより、上記物流シミュレータ 1 a の規模が非常に大きい場合、或いは制約条件が非常に多くて複雑な場合でも、上記物流シミュレータ 1 a に記載された物流状態、数式のうち、スケジュール作成に影響が大きい重要な部分のみを上記数式モデル 3 に取り込むようにすることで、上記物流シミュレータ 1 a の規模を適切な範囲にして、実用的な時間内で最適化計算を行うようにすることができる。

**【0030】**

上記物流シミュレータ 1 a は、考慮すべき物流状態、物流制約を全て記載することができるので、1 回のシミュレーションを行って作成されたスケジュールは現実に実行可能となることが保証される。

**【0031】**

上述したように、本実施形態においては、物流シミュレータ 1 a と、数式モデル 3 と、最適化計算装置 2 とを連動させて物流スケジュールを作成するようにしたので、（1）計算の繰り返しをしないでスケジュールを作成することができる。また、（2）スケジュール作成に影響が大きい重要な部分のみを上記数式モデル 3 に取り込むようにすることで計

算時間を短縮することができるとともに、(3)大規模問題を解くことが可能になる。

【0032】

また、物流指示が必要な事象が発生するたびに上記物流シミュレータ1aの物流状態及び物流制約の情報を検出し、上記検出した検出情報と予め定めた評価指標を元に、上記最適化計算装置2により最適化手法に最適物流指示を計算し、上記計算結果に基づいて上記物流シミュレータ1aで詳細シミュレーションを行ってスケジュールを作成するので、(4)スケジュール精度を高くすることができるとともに、(5)実行可能性の検証が取れているスケジュールを作成することができる。

【0033】

また、数式モデル3を導入したので、スケジュール作成に影響が大きい重要な部分に変更が生じた場合でも迅速に対処することが可能となり、メンテナンス性が高いスケジュール作成装置を構築できる。

【0034】

さらに、シミュレーション制御部1での処理はオンラインで、物流シミュレータ1aに対応させて数学モデル(数式モデル)3の構築や最適化計算装置2での上記数式モデル3に対する最適化計算処理はオフラインで行うように分けることも可能であり、シミュレーション制御部1での処理負荷を軽くして処理能力を高めることができる。

【0035】

なお、本実施形態では、一般的なグラフ理論に基づく例を説明したが、ペトリネットモデルやその他のグラフモデルに本発明を適用することも可能である。ペトリネットモデルでは、上述したグラフモデルにおける点はプレースで表現され、線はトランジションで表現される。トランジションは、プレースからプレースへ製品すなわちトークンを移動させる移動操作端である。

【0036】

一般に、ペトリネットモデルでは、1つのトランジションに複数のプレースからアークが入力している場合には、それらのプレースの全てにトークンが存在していないとトランジションの操作ができない。これをトランジションの発火則と言う。また、1つのトランジションから複数のプレースにアークが出力している場合には、それらのプレースの全てにトークンが出力される。なお、アークとは、トランジションからプレース、或いはプレースからトランジションへトークンが移動する方向を示すものであり、その数は移動するトークンの数を示している。また、トランジションを操作することを、トランジションを発火させると言う。

【0037】

また、グラフモデルでは、各工程内における処理容量をソフトウェア上で制御していたが、ペトリネットモデルでは、各工程内に設けられている仮想プレースに初期設定された容量設定用仮想トークンの数により、各処理工程で同時に処理できる製品の数を制限している。

【0038】

このような特徴を有するペトリネットモデルに本発明を適用する場合も、一連の動作は上述したグラフモデルにおける動作と同様であり、ここでは、その詳細な説明は省略する。

【0039】

(第2の実施形態)

本実施形態の物流スケジュール作成装置も、図1に示したように、物流シミュレータ1aを備えたシミュレーション制御部1、最適化計算装置2等によって構成されている。

【0040】

本実施形態では、物流シミュレータ1aは生産・物流プロセスの物流状態と物流制約を表現した生産・物流プロセスを模擬したシミュレータ、いわば工場を模擬した大型のシミュレータであり、本実施形態においては、ペトリネットモデルを用いてシミュレータを構築するもので、事象(シミュレータのイベント)毎に物を動かす分散系として構成されて

いる。

#### 【0041】

また、上記物流シミュレータ 1 a に対応させて数学モデル（数式モデル）3 が構成されている。本実施形態においては、生産・物流プロセスの物流状態及び物流制約の中から、作成する物流スケジュールに関連する要素を取り込んで、下式(イ)に示すような状態方程式を用いて上記数式モデル 3 が作成されている。上記数式モデル 3 は、半導体記憶手段等により構成される数式モデル保持手段（図示せず）によって保持されている。

$$M(k+1) = M(k) + b \cdot u(k) \cdots (イ)$$

すなわち、離散化したある時刻 k において各ブレースでのトークンの有無を示す状態ベクトル、言い換えれば状態量を  $M(k)$  で表し、各トランジション発火の有無を "1" 及び "0" で表現した操作ベクトルを  $u(k)$  で表わすと、次の時刻 k+1 における状態ベクトル  $M(k+1)$  は、接続行列  $b$  を用いて上式(イ)で表わすことができる。

#### 【0042】

ここで、図 6 にペトリネットモデルの一例を示す。ペトリネットモデルは、ブレース（○印）（図示例では p1~p6）、トランジション（| 印）（図示例では t1~t5）、アーク（→印）、トークン（・印）の 4 要素で表現される。この場合に、1 つのトランジションに複数のブレースからアークが入力している場合には、それらのブレースの全てに製品すなわちトークンが存在しないとトランジションの操作ができない（トランジションの発火則）。また、1 つのトランジションから複数のブレースにアークが出力している場合には、それらのブレースの全てにトークンが出力される。なお、アークとは、トランジションからブレース、或いは、ブレースからトランジションへトークンが移動する方向を示すものであり、その数は移動するトークンの数を示している。また、トランジションを操作することを、トランジションを発火させると言う。

#### 【0043】

図 6 に示すようにトランジション t1 を発火させる場合を例にすると、離散化したある時刻 k において各ブレース p1~p6 でのトークン t1~t5 の有無を示す状態ベクトル、言い換えれば状態量を  $M(k)$  で表し、各トランジション発火の有無を "1" 及び "0" で表現した操作ベクトルを  $u(k)$  で表わすと、次の時刻 k+1 における状態ベクトル  $M(k+1)$  は、接続行列  $b$  を用いて下式 (1) で表わすことができる。

#### 【0044】

【数 1】

$$\begin{matrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \\ p6 \end{matrix} \begin{bmatrix} 0 \\ 1 \\ 3 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{matrix} \begin{bmatrix} 1 \\ 0 \\ 2 \\ 0 \\ 1 \\ 0 \end{bmatrix} \\ M(k) \end{matrix} + \begin{matrix} \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & -2 & 0 \\ 0 & -1 & 1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \\ b \end{matrix} * \begin{matrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ u(k) \end{matrix} = \begin{matrix} \begin{bmatrix} 1 \\ 0 \\ 2 \\ 0 \\ 1 \\ 0 \end{bmatrix} \\ M(k+1) \end{matrix} + \begin{matrix} \begin{bmatrix} -1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ \end{matrix} \cdots (1)$$

#### 【0045】

最適化計算装置 2 では、上記数式モデル 3 に対して最適化計算処理を行い、フィードバックゲイン  $K$  を算出するようにしている。上記最適化計算装置 2 によって行われる最適化計算は線形二次 (LQ) 制御を利用するものであり、下式(ロ)に示すような評価関数  $S$  を用いて行われる。

$$S = \sum \{M' Q M + u' R u\} \cdots (ロ)$$

評価関数  $S$  において、 $Q$ 、 $R$  は制御目的にあわせて設定された適当な行列であり、 $M'$ 、 $u'$  は、それぞれ状態ベクトル  $M$ 、操作ベクトル  $u$  の転置ベクトルである。そして、評価関数  $S$  が最小となるように制御することを考えれば、

$$u(k) = -K \cdot M(k) \cdots (\text{ハ})$$

とした状態フィードバック制御を行うフィードバックゲイン  $K$  を最適制御理論より求めることができる。

#### 【0046】

シミュレーション制御部 1 の最適制御方策部 1 b では、最適化計算装置 2 で算出されるフィードバックゲイン  $K$  と物流状態（状態ベクトル  $M$ ）とを用いて物流指示（操作ベクトル  $u$ ）を算出して物流シミュレータ 1 a に与えて、シミュレーションを進め、新たな物流状態を得て、この新たな物流状態をもとに新たな物流指示を算出するという処理を繰り返す。そして、これにより得られたシミュレーション結果 4 から生産・物流プロセスにおける物流スケジュールを作成する。

#### 【0047】

以下に、第 2 の実施形態の具体例を、図 7～10 を参照しながら説明する。図 7 に示したように、第 1 に、生産・物流プロセスのペトリネットモデルとして、処理時間  $T_p$  に比例した数のプレースで各工程を表したペトリネットモデルを製品種ごとに構築する（ステップ S 4）。

#### 【0048】

第 2 に、構築したペトリネットモデルの状態方程式と、設定した評価関数とからフィードバックゲイン行列  $K$  を製品種ごとに求める（ステップ S 5）。

#### 【0049】

第 3 に、この求めたフィードバックゲイン行列  $K$  と工程内製品仕掛状態を表す状態ベクトル  $M(k)$  とから、製品の移動操作端に対する操作ベクトル  $u(k)$  を製品種ごとに求めたのち、仮想的な生産・物流プロセス内の各移動操作端に対して、正值で大きい操作量が得られた製品順に移動可能な数だけ移動操作を順次行うという方法を用いて、所定期間の生産・物流プロセスのシミュレーションを実行する（ステップ S 6）。

#### 【0050】

図 8 は、図 7 に示したステップ S 4 の処理、すなわち、生産・物流プロセスのペトリネットモデルとして、処理時間  $T_p$  に比例した数のプレースで各工程を表したペトリネットモデルを製品種ごとに構築するための処理手順の例を示すフローチャートである。

#### 【0051】

すなわち、まず離散きざみ時間  $\Delta T$  を入力し（ステップ S 401）、次に製品種ごとに全ての処理工程と処理時間  $T_p$  とを入力する（ステップ S 402）。そして、処理工程ごとに処理時間  $T_p$  をきざみ時間  $\Delta T$  で除して整数化することにより各処理工程のプレース数  $n_p$  を求め（ステップ S 403、S 404）、全処理工程のプレースを連結してペトリネットモデルを製品種ごとに構築する（ステップ S 405）。

#### 【0052】

そして、全ての製品についてペトリネットモデルを構築するまでステップ S 406 からステップ S 402 の処理に戻り、以上の処理を繰り返して行う。なお、離散きざみ時間  $\Delta T$  は、各製品種各工程処理時間の最大公約数をもって定義するのが最も効率的であるが、目的とする制御精度を勘案して適宜設定すればよい。

#### 【0053】

図 9 は、図 7 に示したステップ S 5 の処理、すなわち、構築したペトリネットモデルの状態方程式と設定した評価関数とからフィードバックゲイン行列  $K$  を製品種ごとに求めるための処理手順の例を示すフローチャートである。

#### 【0054】

すなわち、まず製品種ごとにペトリネットモデルの状態方程式を記述する接続行列  $b$  を最適化計算装置 2 に入力するとともに（ステップ S 501）、評価関数を表す行列  $Q$ ,  $R$  を入力する（ステップ S 502）。そして、上記入力した接続行列  $b$  及び評価行列  $Q$ ,  $R$  からフィードバックゲイン行列  $K$  を計算する（ステップ S 503）。この計算を全ての製品について完了するまでステップ S 504 からステップ S 501 の処理に戻り、以上の処理を繰り返して行う。なお、全製品についてペトリネットモデルの接続行列  $b$  を最適化計

算装置 2 に入力し、全製品を一括で計算するようにしてもよい。

#### 【0055】

図 10 は、図 7 に示したステップ S 6 の処理、すなわち、上述のようにして計算したフィードバックゲイン行列  $K$  と工程内製品仕掛状態を表す状態ベクトル  $M(k)$  とから、製品の移動操作端に対する操作ベクトル  $u(k)$  を製品種ごとに求めたのち、仮想的な生産・物流プロセス内の各移動操作端に対して、正值で大きい操作量が得られた製品順に移動可能な数だけ移動操作を順次行うという方法を用いて、所定期間の生産・物流プロセスのシミュレーションを実行するための処理手順の例を示すフローチャートである。

#### 【0056】

すなわち、まずペトリネットモデルの初期状態における状態ベクトル  $M(0)$  を入力するとともに、投入予定の全製品を投入プレースに入力し、時刻  $k$  の値を 0 に初期化したのち（ステップ S 601）、現時点（時刻： $k$ ）における全製品種の状態ベクトル  $M(k)$  を入力する（ステップ S 602）。

#### 【0057】

そして、上述の方法を用いて計算されたフィードバックゲイン行列  $K$  と上記入力した状態ベクトル  $M(k)$  とを掛けて製品種ごとに操作ベクトル  $u(k)$  を計算し（ステップ S 603）、次に移動操作端ごとに、正值で大きい操作量が得られた製品順に、移動可能な数だけ移動操作を実行する（ステップ S 604）。

#### 【0058】

次に、時刻を進めたのち（ステップ S 605）、各ステップのシミュレーションが終了したかどうかを判断し（ステップ S 606）、終了していないときはステップ S 602 の処理に戻る。一方、終了したときは、そのシミュレーション結果をもって生産スケジュールとする（ステップ S 607）。

#### 【0059】

上述したように、本実施形態においては、上述した第 1 の実施形態の効果に加えて、工程内製品仕掛状態を表わす状態ベクトルを各プレースにおける製品の有無に従って表現するようにしたので、フィードバックゲイン行列と上記状態ベクトルから算出される操作ベクトルの各要素を、フィードバックゲイン行列の各要素の大きさをそのまま反映したものとしてことができ、同じプレース内に存在する複数種類の製品のうち、ゲインがより大きいものを常に優先して処理するようにすることができ、その結果、各処理工程での総処理時間を最短にする最適なスケジュールを簡単に作成することができる。

#### 【0060】

また、状態ベクトルを表現する方法として、各プレースにおける製品の有無に従う手法と、各プレースに存在する製品の数に従う手法とのいずれかを選択できるようにすれば、各プレースにおける製品の有無に従って状態ベクトルを表現する場合には、上述のように総処理時間を最短にするスケジュールを簡単に作成することができる一方、各プレースに存在する製品の数に従って状態ベクトルを表現する場合には、1つのプレース内にある製品が溜まってしまうという不都合を防止しつつ総処理時間をある程度短くするスケジュールを簡単に作成することができる。

#### 【0061】

（第 3 の実施形態）

本実施形態の物流スケジュール作成装置も、図 1 に示したように、物流シミュレータ 1 a を備えたシミュレーション制御部 1、最適化計算装置 2 等によって構成されている。

#### 【0062】

本実施形態では、物流シミュレータ 1 a は生産・物流プロセスの物流状態と物流制約を表現した生産・物流プロセスを模擬したシミュレータ、いわば工場を模擬した大型のシミュレータであり、本実施形態においては、ペトリネットモデルを用いてシミュレータを構築するもので、事象（シミュレータのイベント）毎に物を動かす分散系として構成されている。

#### 【0063】



また、上記物流シミュレータ 1 a に対応させて数学モデル（数式モデル）3 が構成されている。本実施形態においては、生産・物流プロセスの物流状態及び物流制約の中から、作成する物流スケジュールに関連する要素を取り込んで、下式(二)に示すような状態方程式を用いて上記数式モデル 3 が作成されている。上記数式モデル 3 は、半導体記憶手段等により構成される数式モデル保持手段（図示せず）によって保持されている。

$$M(k+1) = a \cdot M(k) + b \cdot u(k) \cdots (二)$$

すなわち、離散化したある時刻  $k$  において各ブレースに存在するトークンの数を示す状態ベクトル、言い換えれば状態量を  $M(k)$  で表し、各トランジション発火の有無を "1" 及び "0" で表現した操作ベクトルを  $u(k)$  で表わすと、次の時刻  $k+1$  における状態ベクトル  $M(k+1)$  は、遷移行列  $a$ 、接続行列  $b$  を用いて上式(二)で表わすことができる。

#### 【0064】

最適化計算装置 2 では、上記数式モデル 3 に対して最適化計算処理を行い、フィードバックゲイン  $K$  を算出するようにしている。上記最適化計算装置 2 によって行われる最適化計算は線形二次（LQ）制御を利用するものであり、下式(ロ)に示すような評価関数  $S$  を用いて行われる。

$$S = \sum \{M'QM + u'Ru\} \cdots (ロ)$$

評価関数  $S$  において、 $Q$ 、 $R$  は制御目的にあわせて設定された適当な行列であり、 $M'$ 、 $u'$  は、それぞれ状態ベクトル  $M$ 、操作ベクトル  $u$  の転置ベクトルである。そして、評価関数  $S$  が最小となるように制御することを考えれば、

$$u(k) = -K \cdot M(k) \cdots (ハ)$$

とした状態フィードバック制御を行うフィードバックゲイン  $K$  を最適制御理論より求めることができる。

#### 【0065】

シミュレーション制御部 1 の最適制御方策部 1 b では、最適化計算装置 2 で算出されるフィードバックゲイン  $K$  と物流状態（状態ベクトル  $M$ ）とを用いて物流指示（操作ベクトル  $u$ ）を算出して物流シミュレータ 1 a に与えて、シミュレーションを進め、新たな物流状態を得て、この新たな物流状態をもとに新たな物流指示を算出するという処理を繰り返す。そして、これにより得られたシミュレーション結果 4 から生産・物流プロセスにおける物流スケジュールを作成する。

#### 【0066】

以下に、第 3 の実施形態の具体例を、図 11～14 を参照しながら説明する。図 11 に示したように、第 1 に、生産・物流プロセスのペトリネットモデルとして、処理時間  $T_p$  を入力したブレースで各工程を表したペトリネットモデルを製品種ごとに構築する。そして、構築したペトリネットモデルの各ブレースの処理時間  $T_p$  に比例したきざみ時間遅れ  $n_p$  に従って表される遷移行列  $a$  と接続行列  $b$  とを算出し、これら 2 つの行列  $a$ 、 $b$  を用いて状態方程式を作成する（ステップ S 7）。

#### 【0067】

第 2 に、上記ステップ S 7 で作成した状態方程式と、設定した評価関数とからフィードバックゲイン行列  $K$  を製品種ごとに求める（ステップ S 8）。

#### 【0068】

第 3 に、この求めたフィードバックゲイン行列  $K$  と工程内製品仕掛状態を表す状態ベクトル  $M(k)$  とから、製品の移動操作端に対する操作ベクトル  $u(k)$  を製品種ごとに求めたのち、仮想的な生産・物流プロセス内の各移動操作端に対して、正值で大きい操作量が得られた製品順に移動可能な数だけ移動操作を順次行うという方法を用いて、所定期間の生産・物流プロセスのシミュレーションを実行する（ステップ S 9）。

#### 【0069】

図 12 は、図 11 に示したステップ S 7 の処理、すなわち、生産・物流プロセスのペトリネットモデルとして、処理時間を入力したブレースで各工程を表したペトリネットモデルを製品種ごとに構築し、製品種ごとに各工程でのきざみ時間遅れを表した遷移行列  $a$  と接続行列  $b$  とを算出するための処理手順の例を示すフローチャートである。



**【0070】**

すなわち、まず離散きざみ時間  $\Delta T$  を入力し（ステップ S701）、次に製品種ごとに全ての処理工程と処理時間  $T_p$  とを入力する（ステップ S702）。そして、処理工程ごとに処理時間  $T_p$  をきざみ時間  $\Delta T$  で除して整数化することにより上記各処理工程のきざみ時間遅れ  $n_p$  を求め（ステップ S703、S704）、そのきざみ時間遅れ  $n_p$  に従って、全処理工程の製品種ごとの遷移行列  $a$  と接続行列  $b$  とを算出する（ステップ S705）。

**【0071】**

そして、全ての製品について遷移行列  $a$  と接続行列  $b$  とを算出するまでステップ S706 からステップ S702 の処理に戻り、以上の処理を繰り返して行う。なお、離散きざみ時間  $\Delta T$  は、各製品種各工程処理時間の最大公約数をもって定義するのが最も効率的であるが、目的とする制御精度を勘案して適宜設定すればよい。

**【0072】**

図 13 は、図 11 に示したステップ S8 の処理、すなわち、上記のようにして求めた遷移行列  $a$  及び接続行列  $b$  から成る状態方程式と設定した評価関数とからフィードバックゲイン行列  $K$  を製品種ごとに求めるための処理手順の例を示すフローチャートである。

**【0073】**

すなわち、まず製品種ごとにペトリネットモデルの各プレースの処理時間  $T_p$  に比例したきざみ時間遅れ  $n_p$  を表した遷移行列  $a$  と接続行列  $b$  とを最適化計算装置 2 に入力するとともに（ステップ S801）、評価関数を表す行列  $Q$ 、 $R$  を入力する（ステップ S802）。そして、上記入力した遷移行列  $a$ 、接続行列  $b$  及び評価行列  $Q$ 、 $R$  からフィードバックゲイン行列  $K$  を計算する（ステップ S803）。この計算を全ての製品について完了するまでステップ S804 からステップ S801 の処理に戻り、以上の処理を繰り返して行う。なお、全製品についてペトリネットモデルの各プレースの処理時間  $T_p$  に比例したきざみ時間遅れ  $n_p$  を表した遷移行列  $a$  と接続行列  $b$  とを最適化計算装置 2 に入力し、全製品を一括で計算するようにしてもよい。

**【0074】**

図 14 は、図 11 に示したステップ S9 の処理、すなわち、上述のようにして計算したフィードバックゲイン行列  $K$  と工程内製品仕掛状態を表す状態ベクトル  $M(K)$  とから、製品の移動操作端に対する操作ベクトル  $u(K)$  を製品種ごとに求めたのち、仮想的な生産・物流プロセス内の各移動操作端に対して、正值で大きい操作量が得られた製品順に移動可能な数だけ移動操作を順次行うという方法を用いて、所定期間の生産・物流プロセスのシミュレーションを実行するための処理手順の例を示すフローチャートである。

**【0075】**

すなわち、まずペトリネットモデルの初期状態における状態ベクトル  $M(0)$  を入力するとともに、投入予定の全製品を投入プレースに入力し、時刻  $k$  の値を 0 に初期化したのち（ステップ S901）、現時点（時刻： $k$ ）における全製品種の状態ベクトル  $M(k)$  を入力する（ステップ S902）。

**【0076】**

そして、上述の方法を用いて計算されたフィードバックゲイン行列  $K$  と上記入力した状態ベクトル  $M(k)$  とをかけて製品種ごとに操作ベクトル  $u(k)$  を計算し（ステップ S903）、次に移動操作端ごとに、正值で大きい操作量が得られた製品順に、移動可能な数だけ移動操作を実行する（ステップ S904）。

**【0077】**

次に、時刻を進めたのち（ステップ S905）、各ステップのシミュレーションが終了したかどうかを判断し（ステップ S906）、終了していないときはステップ S902 の処理に戻る。一方、終了したときは、そのシミュレーション結果をもって生産スケジュールとする（ステップ S907）。

**【0078】**

次に、第 3 の実施形態の作用を以下に示す例に即して説明する。ここでは、異なる複数

の工程で複数の製品を処理して製造するプロセスの例として、2種類の製品、すなわち製品Aと製品Bとを4つの工程で処理して製造する場合を考え、各製品に対する各工程の処理時間が表1のように与えられているものとする。

【0079】

【表1】

		工程1	工程2	工程3	工程4
製品A	処理時間(分)	10	15	5	10
	時間きざみ遅れ数	2	3	1	2
製品B	処理時間(分)	10	10	---	10
	時間きざみ遅れ数	2	2	0	2

【0080】

また、離散きざみ時間 $\Delta T$ を5分として両製品A、Bを製造するプロセスのペトリネットモデルを示したものが、図15である。上記した表1は、処理時間を離散きざみ時間 $\Delta T$  (= 5分) で除して得られた各工程1, 2, 3, 4のきざみ時間遅れを表している。この例では、製品Aについては工程1, 2, 3, 4がそれぞれ2, 3, 1, 2のきざみ時間遅れを有し、製品Bについては工程1, 2, 3, 4がそれぞれ2, 2, 0, 2のきざみ時間遅れを有している。この例において製品A、Bは、移動操作端T1 (以降、移動操作端をトランジションとも称する) を経由して投入され、トランジションT6を経由して次工程或いは倉庫等へ搬出される。すなわち、製品AはプレースP1, P2, P3, P4を通り、製品BはプレースP1, P2, P4を通る。

【0081】

上述のように、離散化したある時刻 $k$ において各プレースに存在するトークンの数を示す状態ベクトル、言い換えれば状態量を $M(k)$ で表し、各トランジション発火の有無を"1"及び"0"で表現した操作ベクトルを $u(k)$ で表わすと、次の時刻 $k+1$ における状態ベクトル $M(k+1)$ は、遷移行列 $a$ 、接続行列 $b$ を用いて上式(二)で表わすことができる。状態方程式(二)上の移動操作端(トランジション)と、実プロセスの移動操作端とは一致している。すなわち、操作ベクトル $u(k)$ の信号は、実プロセスの操作信号である。

【0082】

例えば、製品Aに対してプレースP1, P2, P3, P4に存在するトークンの数を並べたベクトルを $M$ 、トランジションT1, T2, T3, T5, T6の発火の有無を"1"及び"0"で表した操作ベクトルを $u$ とすれば、以下の式(2)、(3)で示される遷移行列 $a$ と接続行列 $b$ とを用いて、ベクトル $M(k)$ からベクトル $M(k+1)$ への状態変化を表現することができる。

【0083】

【数 2】

$$a = \begin{array}{c} \begin{array}{cccc} & 1 & 2 & 3 & 4 \\ \begin{array}{c} 00 \\ 11 \end{array} & & & & \\ \begin{array}{c} 000 \\ 100 \\ 011 \end{array} & & & 0 & \\ & & & 1 & \\ 0 & & & 00 & \\ & & & 11 & \end{array} \end{array} \begin{array}{l} 11 \\ 12 \\ 21 \\ 22 \\ 23 \\ 31 \\ 41 \\ 42 \end{array} \quad (2)$$

$$b = \begin{array}{c} \begin{array}{cccccc} & 1 & 2 & 3 & 5 & 6 \\ \begin{array}{c} 10 \\ 0-1 \\ 01 \\ 00 \\ 00-1 \\ 001-1 \\ 0001 \\ 0000-1 \end{array} & & & & & \\ & & & & & \end{array} \end{array} \begin{array}{l} 11 \\ 12 \\ 21 \\ 22 \\ 23 \\ 31 \\ 41 \\ 42 \end{array} \quad (3)$$

【0084】

ただし、上記式 (2) において、列方向の 1 桁の番号はプレース番号を示し、行方向の 2 桁の番号のうち、左側の数字はプレース番号を示し、右側の数字は各プレースに存在する状態 (きざみ時間遅れ) の番号を示している。また、上記式 (3) において、列方向の 1 桁の番号はトランジション番号を示し、行方向の 2 桁の番号のうち、左側の数字はプレース番号を示し、右側の数字は各プレースに存在する状態 (きざみ時間遅れ) の番号を示している。

【0085】

上記遷移行列  $a$  を表現するときの一般形を、次の式 (4) に示す。この式 (4) から明らかなように、ある工程に存在する状態数が  $n$  のとき、その工程に関する部分の小行列は、式 (4) 中に示したような  $n \times n$  の正方行列で表される。そして、全工程の遷移を表した遷移行列  $a$  は、上記各工程に関する小行列を行方向及び列方向の番号に対応した適当な位置に配置するとともに、その他の要素を全て "0" とすることによって表される。

【0086】

## 【数 3】

$$a = \begin{matrix} & \begin{matrix} 1 & 2 & & & n-1 & n \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ \vdots \\ n-1 \\ n \end{matrix} & \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix} \end{matrix} \quad \begin{matrix} 1: n \text{ 時間遅れの表現} \\ 2 \\ (4) \\ n-1 \\ n \end{matrix}$$

状態数 1 の場合の小行列  
状態数 2 の場合の小行列  
状態数 3 の場合の小行列

## 【0087】

また、上記接続行列  $b$  を表現するときの一般形を、次の式 (5) に示す。すなわち、ある工程に存在する状態数が  $n$  のとき、その工程に関する部分の小行列は、式 (5) に示すような  $n \times 2$  の行列で表される。(1,1) 要素の数 "1" は、該当するプレースにトークンが入力されることを示し、(n,2) 要素の数 "-1" は、該当するプレースからトークンが出力されることを示している。そして、全工程に関する接続行列  $b$  は、上記各工程の小行列を行方向及び列方向の番号に対応した適当な位置に配置するとともに、その他の要素を全て "0" とすることによって表される。

## 【0088】

## 【数 4】

$$b = \begin{matrix} & \begin{matrix} 1 & 0 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ \vdots \\ n-1 \\ n \end{matrix} & \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \\ 0 & -1 \end{pmatrix} \end{matrix} \quad \begin{matrix} 1: n \text{ 時間遅れの表現} \\ 2 \\ (5) \\ n-1 \\ n \end{matrix}$$

## 【0089】

製品 A について上記式 (2)、(3) で示される遷移行列  $a$  と接続行列  $b$  とを用いてベクトル  $M(k)$  からベクトル  $M(k+1)$  への状態変化を表現したのと同様のことが製品 B についても表現できる。

## 【0090】

このように、製品ごとに、処理時間を入力したプレースを持つペトリネットモデルを用い、製品ごとに各工程でのきざみ時間遅れを表した遷移行列  $a$  と接続行列  $b$  とで各工程を表すことにより、離散時刻  $k$  が 1 進むたびに複数の離散きざみ時間を要するプレースでトークンが移動していくことを表現でき、その結果、離散時間系の最適制御理論を適用できる形にすることができる。

## 【0091】

この場合、1つのプレースに複数単位の処理時間（離散きざみ時間）を割り当てるようにしてペトリネットモデルを構築し、遷移行列  $a$  及び接続行列  $b$  の 2つの行列式だけで物流モデルを表現できるようにしているのので、1つの離散きざみ時間ごとに1つのプレースを設けた制御用のペトリネットモデルを再構築しなくても済む。また、上記1つの離散きざみ時間ごとにプレースを設けたペトリネットモデルに比べて制御端の数を少なくするこ

とができ、実際のプロセスに使いやすいモデルとすることができる。

#### 【0092】

これにより、制御の目的にあわせて適当な行列  $Q$ ,  $R$  を設定して、評価関数 ( $J$ ) が最小になるように制御することを考えれば、上式 (ハ) とした状態フィードバック制御を行うフィードバックゲイン行列  $K$  を最適制御理論より求めることができる。フィードバックゲイン行列  $K$  の計算法としては、例えば (安藤和昭、他編著「数値解析手法による制御系設計」計測自動制御学会発行、pp.126-130、平成5年初版第2刷) に記載されたいくつかの方法がある。

#### 【0093】

また、評価行列  $Q$ ,  $R$  は、それぞれプレースの数及びトランジションの数の次元を有する正方行列で、それぞれ制御の過渡特性とトランジション操作の入力エネルギーとを評価関数として表すためのものであり、制御目的に応じて適宜設定すればよい。例えば、過渡特性を重視する場合は、大きな数値をもつ行列  $Q$  を設定すればよい。また、操作に要する入力エネルギーを小さく抑えたい場合は、大きな数値をもつ行列  $R$  を設定すればよい。

#### 【0094】

次に、第3の一実施形態の具体例を、モデル構築、制御則構築、シミュレーション、実施結果例に分けて以下に説明する。

#### 〔モデル構築〕

図16は、複数工程よりなる製造プロセスのペトリネットモデルの一例を示したものであり、離散きざみ時間  $\Delta T$  は5分に設定してある。この製造プロセスは、工程1, 2, 3, 4 からなる前処理工程と工程5, 6 からなる後処理工程とから構成され、プレース  $P1$  (投入プレース) で示した前処理工程入側バッファと、プレース  $P6, P8$  で示した後処理工程入側バッファと、プレース  $P10$  で示した搬出用プレースとを具備している。

#### 【0095】

この製造プロセスでは、前処理工程1又は2で処理された製品は後処理工程5で処理され、また、前処理工程3又は4で処理された製品は後処理工程6で処理されるが、製品によっては前処理工程1～4と後処理工程5～6の複数の組合せのいずれかで処理し得るものがあるため、全体で15の製品種類がある。次の表2は、それぞれの製品種がどの前処理工程と後処理工程との組合せで処理可能かを示したものである。

#### 【0096】

【表2】

製品種No	(前処理工程No., 後処理工程No.)	製品種No	(前処理工程No., 後処理工程No.)
1	(1,5)	9	(2,5) (4,6)
2	(2,5)	10	(3,6) (4,6)
3	(3,6)	11	(1,5) (2,5) (3,6)
4	(4,6)	12	(1,5) (2,5) (4,6)
5	(1,5) (2,5)	13	(1,5) (3,6) (4,6)
6	(1,5) (3,6)	14	(2,5) (3,6) (4,6)
7	(1,5) (4,6)	15	(1,5) (2,5) (3,6) (4,6)
8	(2,5) (3,6)		

#### 【0097】

図16において、プレース  $P11, P12, P13, P14, P15, P16$  は、それぞれ工程1～6の処理容量を制限するための仮想プレースであり、これらの仮想プレースに初期設定された容量設定用仮想トークンの数により、各処理工程で同時に処理できる製品の数を制限している。例えば、工程1は、プレース  $P2$  から構成されるとともに、仮想プレース  $P11$  に初期状態と

して1個の容量設定用仮想トークンが置かれることにより、この工程1での処理容量が1に設定されている。

#### 【0098】

すなわち、工程1に製品がなかった状態からトランジションT2を経由して製品が入力されるときに、仮想プレースP11に置かれた容量設定用仮想トークンが同時に入力され、製品を表すトークンがプレースP1からプレースP2に移動する。その後、そのトークンがトランジションT3を経由してプレースP6に移動すると同時に容量設定用仮想トークンがプレースP11に復帰して、再び工程1が待ち状態になる。

#### 【0099】

このように、複数のプレースで表される工程に対して仮想プレースを適宜利用することにより、各工程の処理容量を設定して製造プロセスのペトリネットモデルを構築することは周知の手法である。

#### 【0100】

なお、この図16の例は、工程1から工程6までの仮想プレースP11, P12, P13, P14, P15, P16の容量が全て1のプロセスである。また、製品入力用のプレースP1と製品出力用のプレースP10は最大200個の製品を収容可能で、プレースP6, P8のバッファはともに最大20個の製品を収容可能と設定してある。

#### 【0101】

図17に実線で示された部分は、表2に記載した製品種No.1に対するペトリネットモデルを示したものである。同様に、全ての製品種に対してもペトリネットモデルを構築することにより、図11のフローチャートに示したモデル構築が行われる。

#### 【0102】

##### 〔制御則構築〕

引き続き製品種No.1を例にとり、フィードバックゲイン行列Kを求める過程を説明する。式(二)の状態方程式における遷移行列aは、図17に示された製品種No.1のペトリネットモデルに対しては、次の式(6)で表現されるサイズ8×8の行列として求められる。ただし、式(6)において、列方向の番号はプレース番号を示し、行方向の番号のうち、下1桁以外の数字はプレース番号を示し、下1桁の数字は各プレースに存在する状態(きざみ時間遅れ)の番号を示している。

#### 【0103】

##### 【数5】

$$a = \begin{matrix} & \begin{matrix} 1 & 2 & 6 & 7 & 10 \end{matrix} \\ \begin{matrix} 1 \\ 00 \\ 11 \\ \\ \\ 0 \end{matrix} & \begin{bmatrix} 1 & & & & \\ & 0 & & 0 & \\ & 1 & 1 & & \\ & & 1 & & \\ & & & 0 & 0 & 0 \\ & & & 1 & 0 & 0 \\ & & & 0 & 1 & 1 \\ & & & & & 1 \end{bmatrix} & \begin{matrix} 11 \\ 21 \\ 22 \\ 61 \\ 71 \\ 72 \\ 73 \\ 101 \end{matrix} \end{matrix} \quad (6)$$

#### 【0104】

また、式(二)の状態方程式における接続行列bは、図17に示された製品種No.1のペトリネットモデルに対しては、次の式(7)で表現されるサイズ8×6の行列として求められる。ただし、式(7)において、列方向の番号はトランジション番号を示し、行方向の番号のうち、下1桁以外の数字はプレース番号を示し、下1桁の数字は各プレースに存在する状態の番号を示している。

#### 【0105】

## 【数 6】

$$b = \begin{matrix} & 1 & 2 & 3 & 10 & 11 & 14 \\ \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} & \begin{matrix} 11 \\ 21 \\ 22 \\ 61 \\ 71 \\ 72 \\ 73 \\ 101 \end{matrix} \end{matrix} \quad (7)$$

## 【0106】

また、式(ロ)で表される評価関数中における行列Q, Rは、上述したように、それぞれ各工程のきざみ時間遅れ数及びトランジションの数の次元を有する正方行列であるから、それぞれペトリネットモデルの製品種ごとに設定される。

## 【0107】

製品種No.1に対しては、この製造プロセスにおける標準的な値として、

$$Q = 20 \times I(8) \quad \dots (8)$$

及び

$$R = I(6) \quad \dots (9)$$

を設定した。ただし、式(8)、(9)においてI(n)はn次の単位行列である。

## 【0108】

このようにして遷移行列a及び接続行列bと、評価関数を定義する行列Q, Rとが定まれば、離散時間系における最適制御理論によりフィードバックゲイン行列Kを求めることができる。

## 【0109】

すなわち、代数リッカチ方程式(10)

$$P = A^T P A - A^T P B (B^T P B + R)^{-1} B^T P A + Q \quad \dots (10)$$

を満足する解Pを求めれば、フィードバックゲイン行列Kは、

$$K = -(B^T P B + R)^{-1} B^T P A \quad \dots (11)$$

により計算することができる。ここで、 $(B^T P B + R)^{-1}$ は行列 $(B^T P B + R)$ の逆行列である。なお、リッカチ方程式の解法は、上述した参考文献にも詳述されているように既知の方法が多数知られている。

## 【0110】

このようにして計算した製品種No.1のペトリネットモデルに対するフィードバックゲイン行列Kは、次の式(12)で与えられる。ただし、この式(12)において、列方向の番号及び行方向の番号は、それぞれ式(7)における行方向の番号及び列方向の番号に対応するものである。

## 【0111】

## 【数 7】

$$K = - \begin{matrix} & 11 & 21 & 22 & 61 & 71 & 72 & 73 & 101 \\ \begin{pmatrix} -0.927 & -0.252 & -0.252 & -0.249 & -0.001 & 0.000 & 0.000 & 0.000 \\ 0.029 & -0.259 & -0.259 & -0.255 & -0.001 & 0.000 & 0.000 & 0.000 \\ 0.003 & 0.629 & 0.629 & -0.343 & -0.001 & 0.000 & 0.000 & 0.000 \\ 0.006 & 0.292 & 0.292 & 0.300 & -0.003 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.002 & 0.002 & 0.002 & 0.073 & 0.919 & 0.919 & -0.039 \\ 0.000 & 0.004 & 0.004 & 0.004 & 0.107 & 0.880 & 0.880 & 0.919 \end{pmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 10 \\ 11 \\ 14 \end{matrix} \end{matrix} \quad (12)$$

## 【0112】

同様にして全ての製品種に対してもフィードバックゲイン行列  $K$  を計算することにより、図 11 のフローチャートに示した制御則の構築が行われる。

#### 【0113】

〔シミュレーション〕

次に、シミュレーションについて、図 14 に示したフローチャートに沿って説明する。まず、ペトリネットモデルの初期状態における状態ベクトル  $M(0)$  を入力するとともに、投入予定の全製品を投入プレースに入力し、時刻  $k$  の値を 0 に初期化したのち（ステップ S901）、現時点（時刻： $k$ ）における全製品種の状態ベクトルを入力し（ステップ S902）、新規投入製品を製品種ごとに入力して状態ベクトルに加えると、現時点における最終的な状態ベクトル  $M(k)$  が定まる。さらに、製品種ごとに、フィードバックゲイン行列  $K$  と状態ベクトル  $M(k)$  とをかければ、操作ベクトル  $u(k)$  が計算できる（ステップ S903）。

#### 【0114】

例えば、ある時刻  $k$  において製品種 No.1 のトークンがプレース P6 のみに 1 個あり、さらに同品種の新規投入製品は無かったとすると、製品種 No.1 に対する最終的な状態ベクトル  $M(k)$  は、

$$M(k) = [0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]' \cdots (13)$$

となる。ただし、式 (13) において記号' は転置ベクトルであることを表す。また、式 (13) の各要素は、それぞれ各プレース P1, P2, P6, P7, P10 の状態番号 11, 21, 22, 61, 71, 72, 73, 101 に対応する製品種 No.1 のトークン数を表す。

#### 【0115】

次に、式 (12) で与えられたフィードバックゲイン行列  $K$  と式 (13) の状態ベクトル  $M(k)$  とを式 (8) に従ってかければ、操作ベクトル  $u(k)$  は、

$$u(k) = [-0.249\ -0.255\ -0.3430\ 0.300\ 0.002\ 0.004]' \cdots (14)$$

と求められる。ただし、式 (14) の各要素は、それぞれトランジション T1, T2, T3, T10, T11, T14 に対する操作量である。

#### 【0116】

同様にして全ての製品種に対しても操作ベクトル  $u(k)$  を求める。そして、移動操作端すなわちトランジションごとに、正值で大きい操作量が得られた製品順に、移動可能な数だけ移動操作を実行する（ステップ S904）。

#### 【0117】

例えば、時刻  $k$  において製品種 No.1 のトークンがプレース P6 に 1 個あり、このとき同時に製品種 No.8 のトークンもプレース P6 に 1 個あり、他の製品種のトークンはプレース P6 にはなかったとする。この場合、トランジション T10 に関して製品種 No.1 のトークンに対する操作量は、式 (14) より 0.300 であった。一方、製品種 No.8 のトークンに対する操作量を同様に計算する。製品種 No.8 の遷移行列  $a$  を式 (15)、接続行列  $b$  を (16)、行列  $Q$ 、 $R$  を (17)、(18) を示す。フィードバックゲイン行列  $K$  は次の式 (19) のようになり、トランジション T10 の操作量は 0.361 となる。

#### 【0118】



## 【数 8】

【 0 1 1 9 】

## 【数 9】

[illegible]

【 0 1 2 0 】

【数 1 0】

$$Q = 20 \times I(16) : I(n) \text{ は } n \text{ 次の単位行列} \quad (17)$$

**【 0 1 2 1 】**

【数 1 1】

$$R = I(10) : I(n) \text{ は } n \text{ 次の単位行列} \quad (18)$$

**【 0 1 2 2 】**

## 【数 12】

$$K = \begin{pmatrix} 11 & 31 & 32 & 33 & 41 & 42 & 43 & 44 & 61 & 71 & 72 & 73 & 81 & 91 & 92 & 101 \\ -0.925 & -0.131 & -0.107 & -0.107 & -0.056 & -0.033 & -0.023 & -0.023 & -0.105 & -0.001 & 0.000 & 0.000 & -0.023 & 0.000 & 0.000 & 0.000 & 1 \\ 0.018 & -0.134 & -0.109 & -0.109 & -0.001 & -0.001 & -0.001 & -0.001 & -0.103 & 0.000 & 0.000 & 0.000 & -0.001 & 0.000 & 0.000 & 0.000 & 4 \\ 0.001 & 0.116 & 0.663 & 0.663 & 0.000 & -0.002 & -0.001 & -0.001 & -0.310 & -0.001 & 0.000 & 0.000 & -0.001 & 0.000 & 0.000 & 0.000 & 5 \\ 0.013 & 0.000 & 0.000 & 0.000 & -0.056 & -0.032 & -0.023 & -0.023 & 0.000 & 0.001 & 0.000 & 0.000 & -0.022 & 0.000 & 0.000 & 0.000 & 6 \\ 0.000 & 0.001 & 0.002 & 0.002 & 0.065 & 0.137 & 0.721 & 0.721 & 0.002 & -0.007 & -0.002 & -0.002 & -0.250 & -0.002 & -0.002 & -0.001 & 7 \\ 0.002 & 0.208 & 0.354 & 0.354 & -0.001 & -0.003 & -0.003 & -0.003 & 0.361 & -0.003 & 0.000 & 0.000 & -0.003 & 0.000 & 0.000 & 0.000 & 10 \\ 0.000 & 0.002 & 0.003 & 0.003 & 0.005 & 0.009 & 0.014 & 0.014 & 0.003 & 0.070 & 0.921 & 0.921 & 0.014 & -0.033 & -0.033 & -0.036 & 11 \\ 0.001 & 0.003 & 0.003 & 0.003 & 0.127 & 0.249 & 0.471 & 0.471 & 0.003 & -0.014 & -0.003 & -0.003 & 0.482 & -0.004 & -0.004 & -0.003 & 12 \\ 0.000 & 0.002 & 0.002 & 0.002 & 0.011 & 0.020 & 0.035 & 0.035 & 0.002 & 0.026 & -0.033 & -0.033 & 0.036 & 0.921 & 0.921 & -0.036 & 13 \\ 0.000 & 0.005 & 0.007 & 0.007 & 0.020 & 0.038 & 0.065 & 0.065 & 0.007 & 0.126 & 0.851 & 0.851 & 0.066 & 0.851 & 0.851 & 0.888 & 14 \end{pmatrix} \quad (19)$$

## 【0123】

ここで、トランジションT10は、移動操作すなわち発火によって1個のトークンをプレースP7に移動させることができるものであるとする。この場合、時刻kにおいては仮想プレースP15に容量設定用仮想トークンが1個存在しているため、結局製品種No. 8のトークンに対してのみトランジションT10を発火させることが可能になる。また、製品種No. 8のトークンは時刻kにおいて他のプレースには存在していなかったから、操作量の正負に関わらずトランジションT10以外のトランジションは発火できない。

## 【0124】

したがって、製品種No. 8に対する最終的な操作ベクトルu(k)は、

$$u(k) = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]' \cdots (20)$$

となり、トランジションT10のみが発火する。その結果、製品種No. 8に対する時刻kにおける状態ベクトルが

$$M(k) = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]' \cdots (21)$$

から時刻k+1における状態ベクトルへ

$$M(k+1) = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]' \cdots (22)$$

と更新される。

## 【0125】

同様の計算を全製品種、全トランジションについて行えば、全製品種に対する時刻k+1における状態ベクトルM(k+1)が計算されることになる。その後は、時間を進めたのち(ステップS905)、各ステップのシミュレーションが終了したかどうかを判断し(ステップS906)、終了していないときはステップS902の処理に戻る。一方、終了したときは、そのシミュレーション結果をもって生産スケジュールとする(ステップS907)。

## 【0126】

上述したように、本実施形態においては、上述した第1の実施形態の効果に加えて、1つのプレースに複数のきざみ時間遅れを含むようにしてペトリネットモデルを構築し、遷移行列及び接続行列の2つの行列式だけで数式モデルを表現できるようにしたので、1つのきざみ時間遅れごとに1つのプレースを設けたペトリネットモデルを再構築しなくても済む。また、1つのきざみ時間遅れごとに1つのプレースを設けたペトリネットモデルに比べて制御端の数を少なくすることができ、実プロセスに使いやすいものとすることができる。

## 【0127】

なお、本実施形態では、ペトリネットモデルを例に挙げて説明したが、その他のグラフモデル、例えば有向グラフや無向グラフに本発明を適用することも可能である。例えば有向グラフでは、上述したペトリネットモデルにおけるプレースは点で表現され、トランジションは矢印付きの線で表現される。矢印付きの線は、点から点へ製品すなわちトークンを移動させる移動操作端であり、移動路の役目をする。また、矢印は、線から点、或いは

、点から線へトークンが移動する方向を示すものである。

【0128】

また、ペトリネットモデルでは、各工程内に設けられている仮想プレースに初期設定された容量設定用仮想トークンの数により、各処理工程で同時に処理できる製品の数制限していたが、通常のグラフモデルでは仮想プレースは存在せず、各工程内における処理容量をソフトウェア上で制御される。

【0129】

このような特徴を有する通常のグラフモデルに本発明を適用する場合も、一連の動作は上述したペトリネットモデルにおける動作と同様であり、ここでは、その詳細な説明は省略する。

【0130】

図18は、上述した生産・物流スケジュール作成装置を作成可能なコンピュータシステムの一例を示すブロック図である。同図において、1200はコンピュータPCである。PC1200は、CPU1201を備え、ROM1202又はハードディスク（HD）1211に記憶された、或いはフレキシブルディスクドライブ（FD）1212より供給されるデバイス制御ソフトウェアを実行し、システムバス1204に接続される各デバイスを総括的に制御する。

【0131】

上記PC1200のCPU1201、ROM1202又はハードディスク（HD）1211に記憶されたプログラムにより、本実施形態の各機能手段が構成される。

【0132】

1203はRAMで、CPU1201の主メモリ、ワークエリア等として機能する。1205はキーボードコントローラ（KBC）であり、キーボード（KB）1209から入力される信号をシステム本体内に入力する制御を行う。1206は表示コントローラ（CRTC）であり、表示装置（CRT）1210上の表示制御を行う。1207はディスクコントローラ（DKC）で、ブートプログラム（起動プログラム：パソコンのハードやソフトの実行（動作）を開始するプログラム）、複数のアプリケーション、編集ファイル、ユーザファイルそしてネットワーク管理プログラム等を記憶するハードディスク（HD）1211、及びフレキシブルディスク（FD）1212とのアクセスを制御する。

【0133】

1208はネットワークインタフェースカード（NIC）で、LAN1220を介して、ネットワークプリンタ、他のネットワーク機器、或いは他のPCと双方向のデータのやり取りを行う。

【0134】

なお、本発明は複数の機器から構成されるシステムに適用しても、一つの機器からなる装置に適用してもよい。

【0135】

また、本発明の目的は前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システム或いは装置に供給し、そのシステム或いは装置のコンピュータ（CPU若しくはMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。

【0136】

この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0137】

プログラムコードを供給するための記憶媒体としては、例えばフレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモ리카ード、ROM等を用いることができる。

【0138】

また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）等が実際の処理の一部又は全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

#### 【0139】

更に、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPU等が実際の処理の一部又は全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

#### 【0140】

以上述べた実施形態では、シミュレーション結果から上記生産・物流プロセスにおける生産・物流スケジュールを作成する例を説明したが、シミュレーション結果に基づいて生産・物流プロセスの制御を行う場合にも本発明は適用される。

#### 【図面の簡単な説明】

#### 【0141】

【図1】本発明を適用した生産・物流スケジュール作成装置の概略構成を説明するブロック図である。

【図2】第1の実施形態における生産・物流スケジュール作成の処理手順を示すフローチャートである。

【図3】モデル構築の処理手順を示すフローチャートである。

【図4】制御則構築の処理手順を示すフローチャートである。

【図5】シミュレーションの処理手順を示すフローチャートである。

【図6】ペトリネットモデルの例を示す図である。

【図7】第2の実施形態における生産・物流スケジュール作成の処理手順を示すフローチャートである。

【図8】モデル構築の処理手順を示すフローチャートである。

【図9】制御則構築の処理手順を示すフローチャートである。

【図10】シミュレーションの処理手順を示すフローチャートである。

【図11】第3の実施形態における生産・物流スケジュール作成の処理手順を示すフローチャートである。

【図12】モデル構築の処理手順を示すフローチャートである。

【図13】制御則構築の処理手順を示すフローチャートである。

【図14】シミュレーションの処理手順を示すフローチャートである。

【図15】ペトリネットモデルの例を示す図である。

【図16】ペトリネットモデルの例を示す図である。

【図17】製品種No.1に対するペトリネットモデルの例を示す図である。

【図18】本発明の生産・物流スケジュール作成装置を構成可能なコンピュータシステムの一例を示すブロック図である。

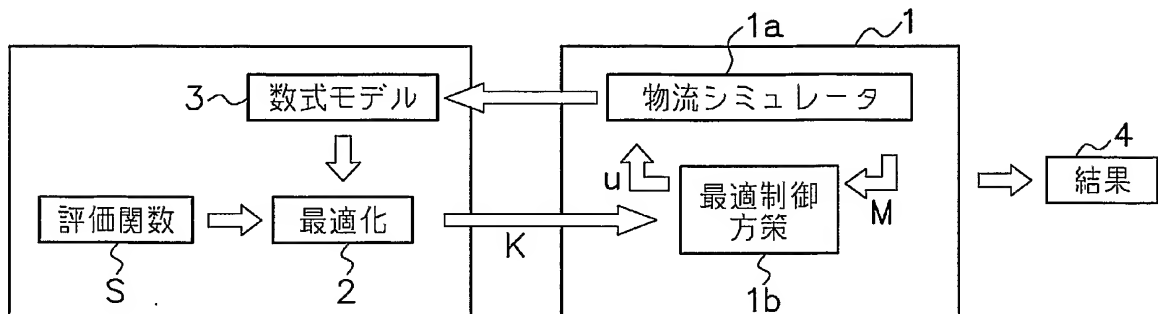
#### 【符号の説明】

#### 【0142】

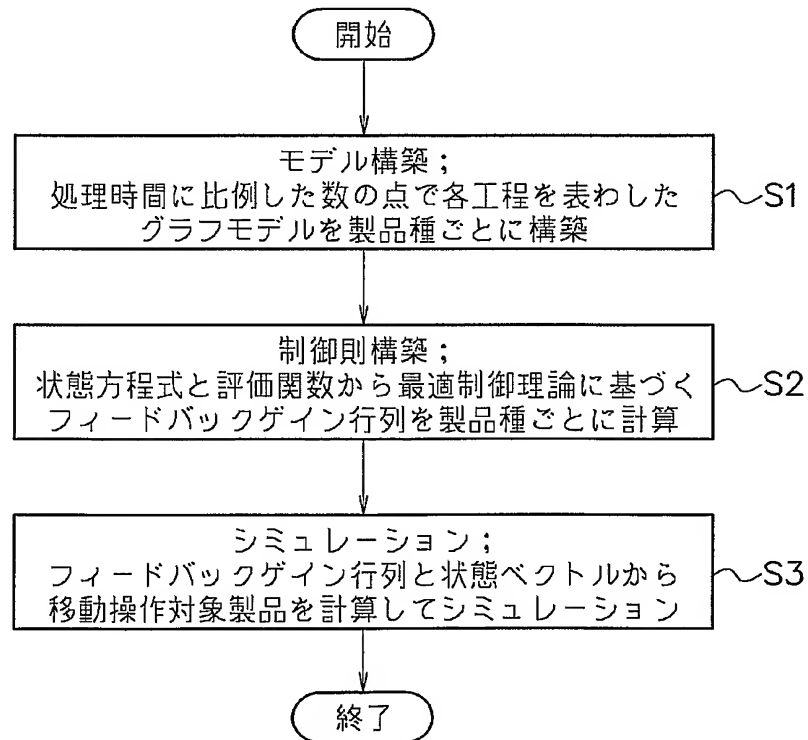
- 1 シミュレーション制御部
- 1 a 物流シミュレータ
- 1 b 最適制御方策部
- 2 最適化計算装置
- 3 数式モデル

【書類名】 図面

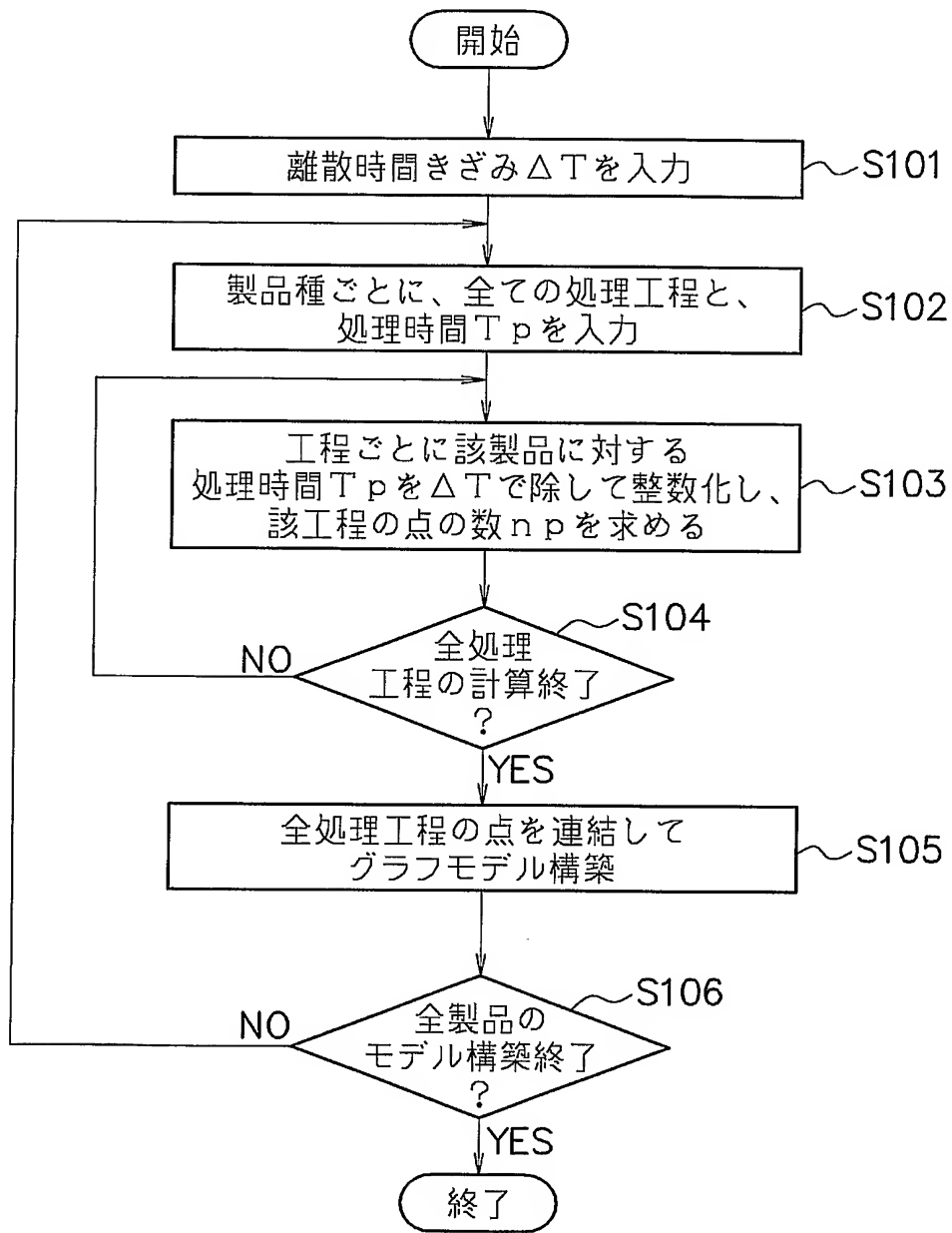
【図 1】



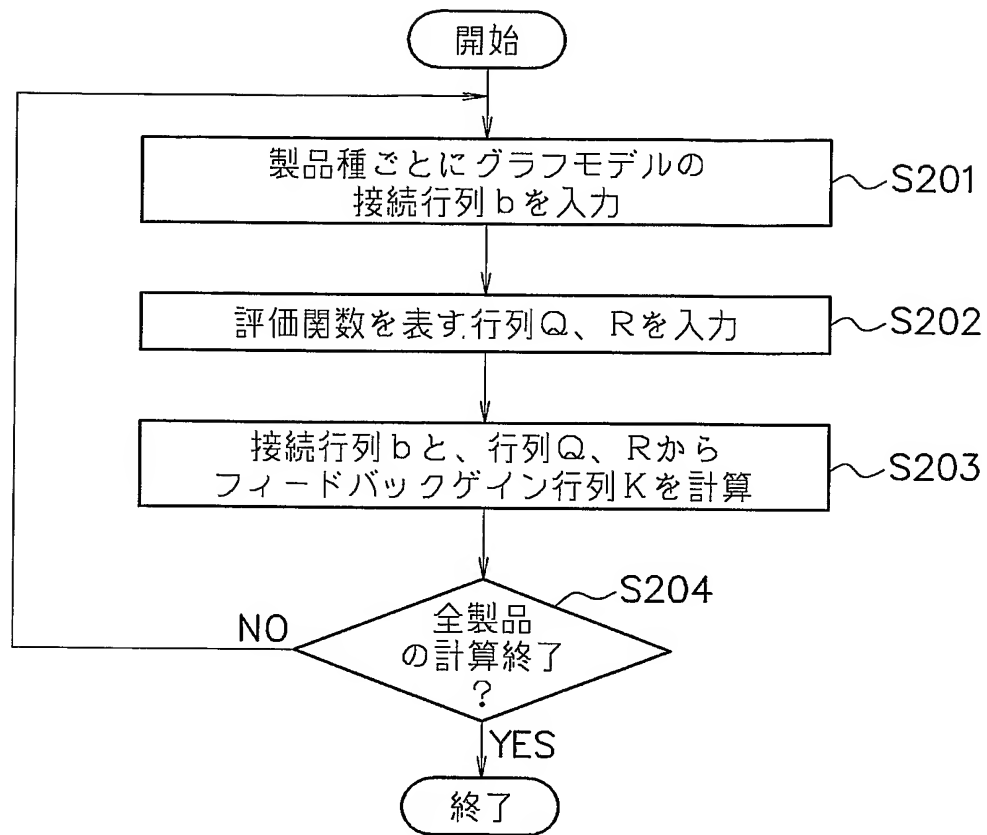
【図 2】



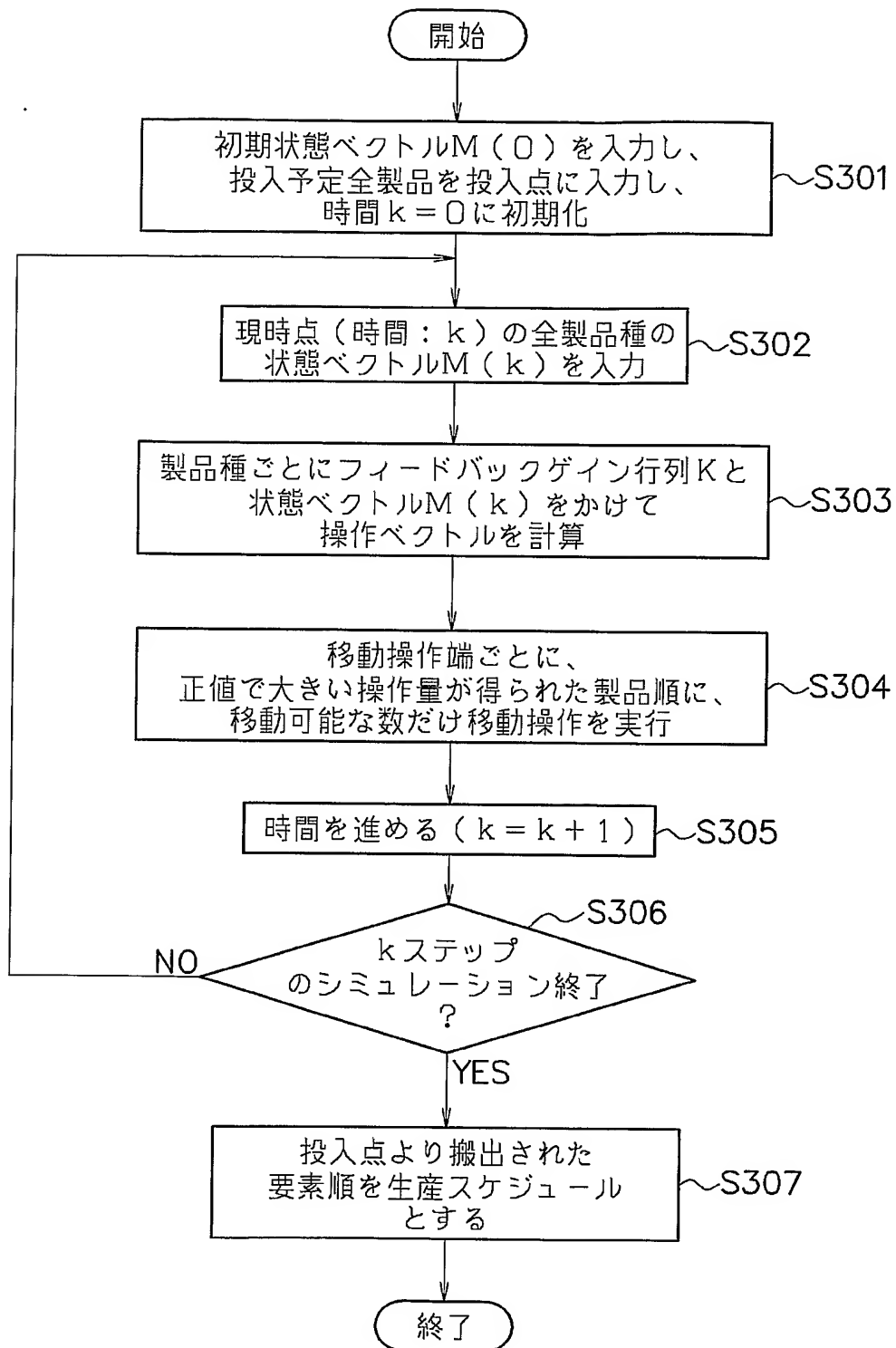
【図 3】



【図 4】

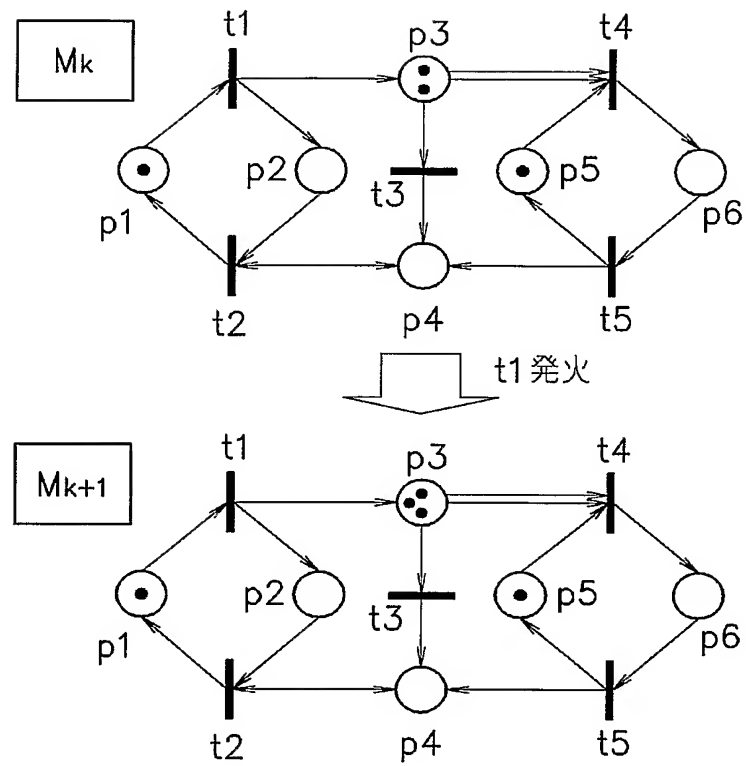


【図 5】

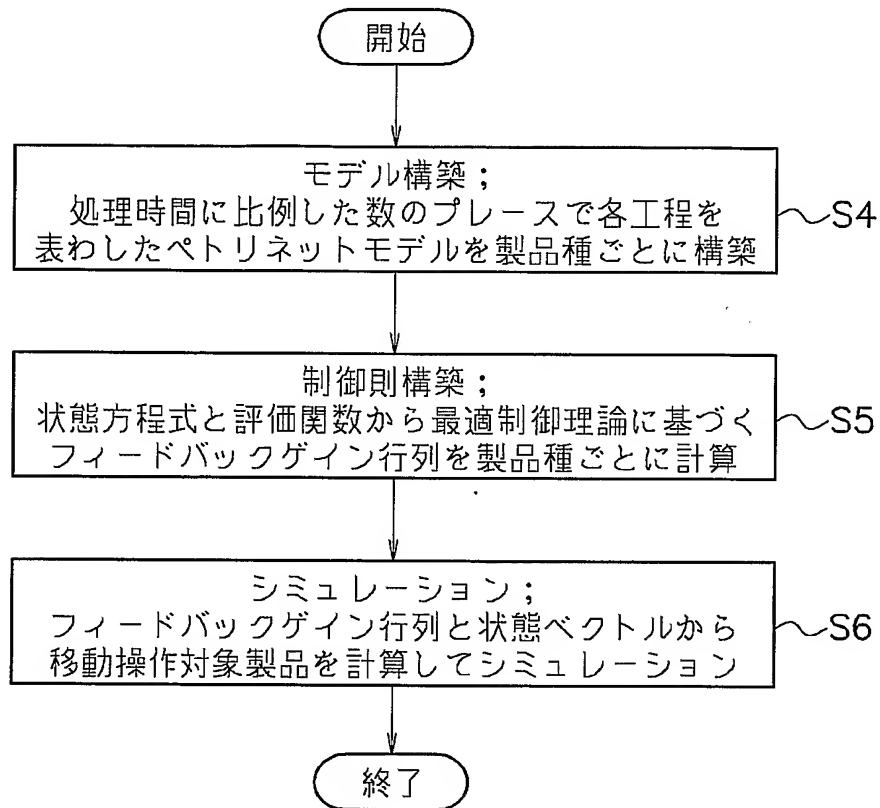




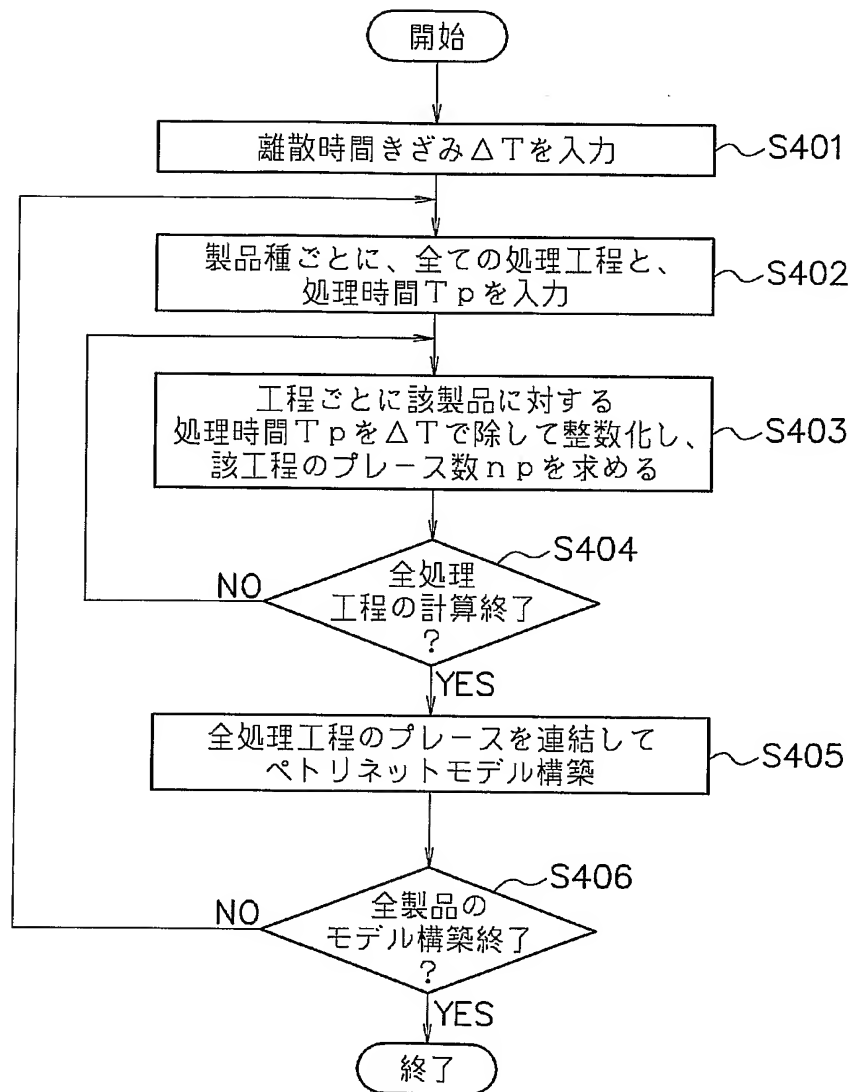
【図 6】



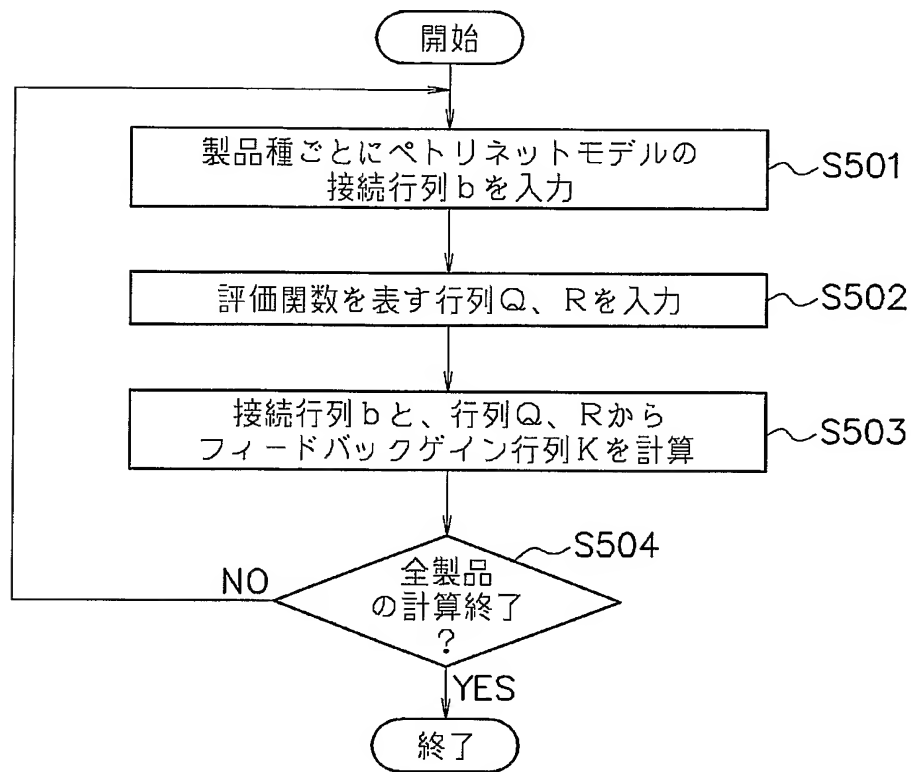
【図 7】



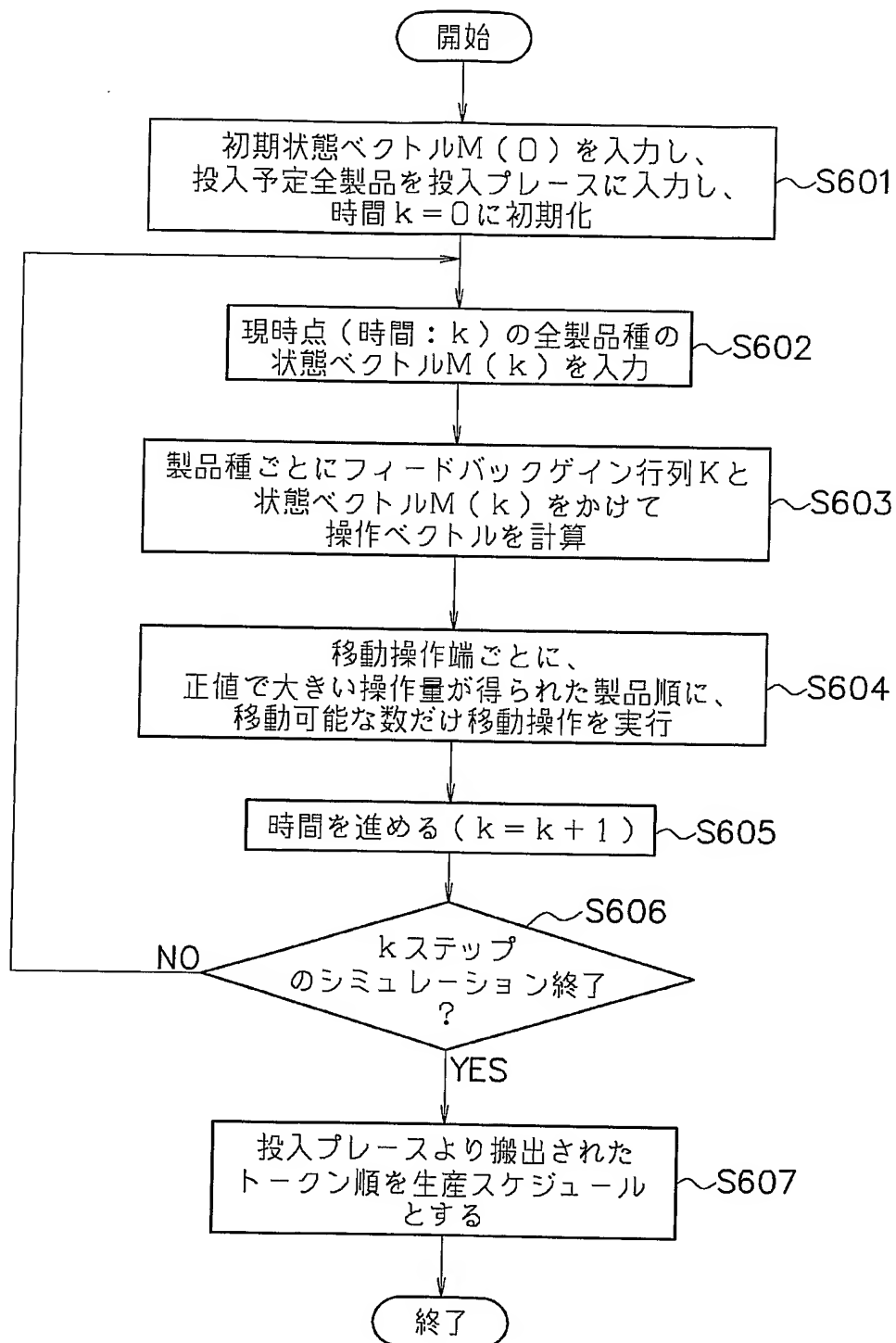
【図 8】



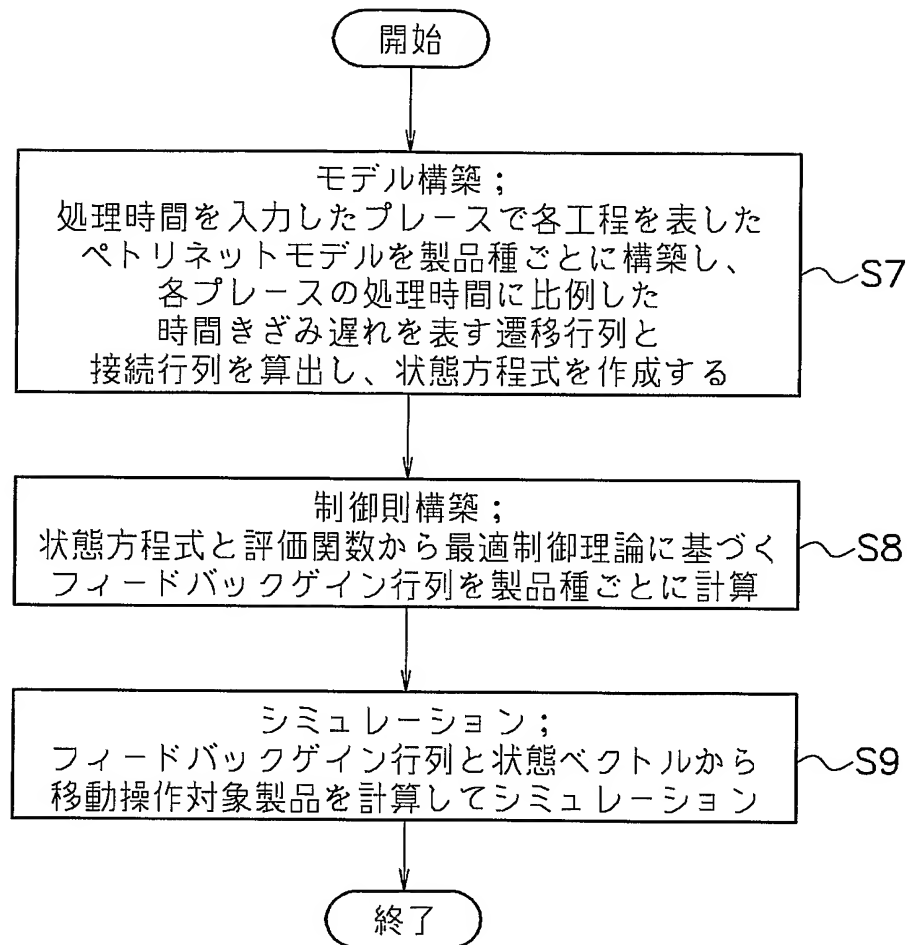
【図 9】



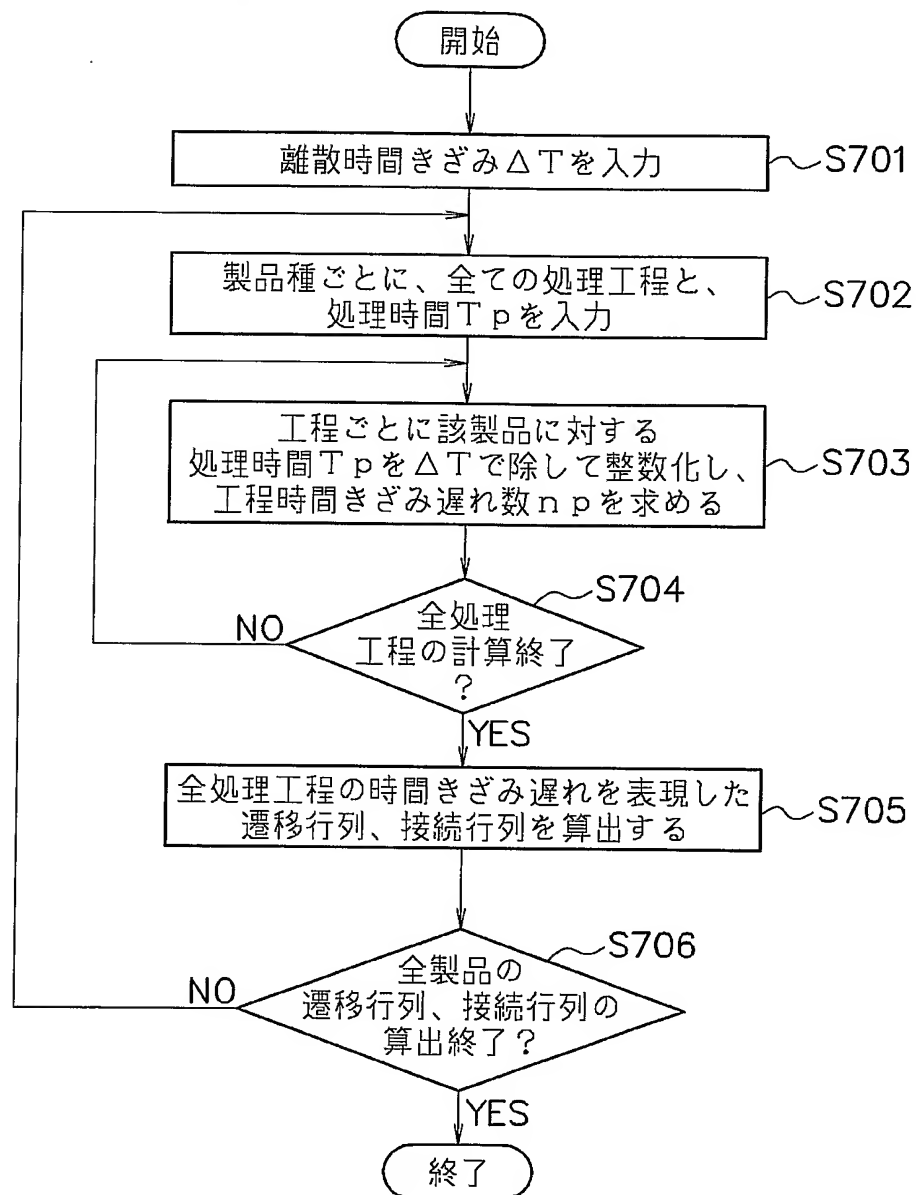
【図 10】



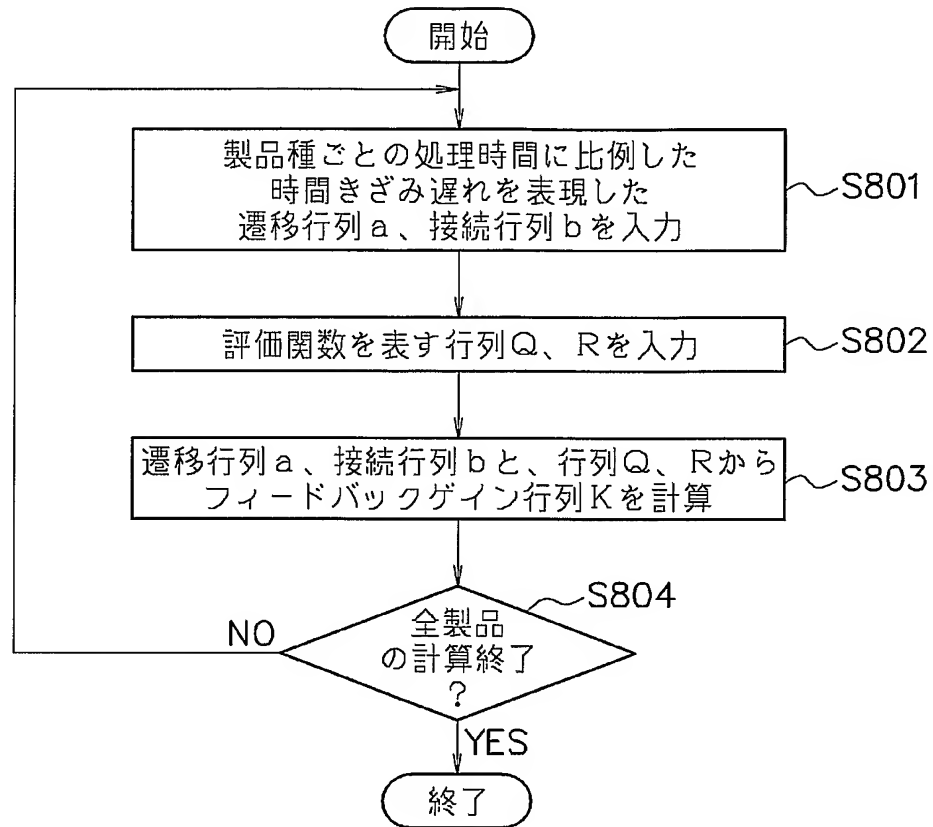
【図 11】



【図 12】

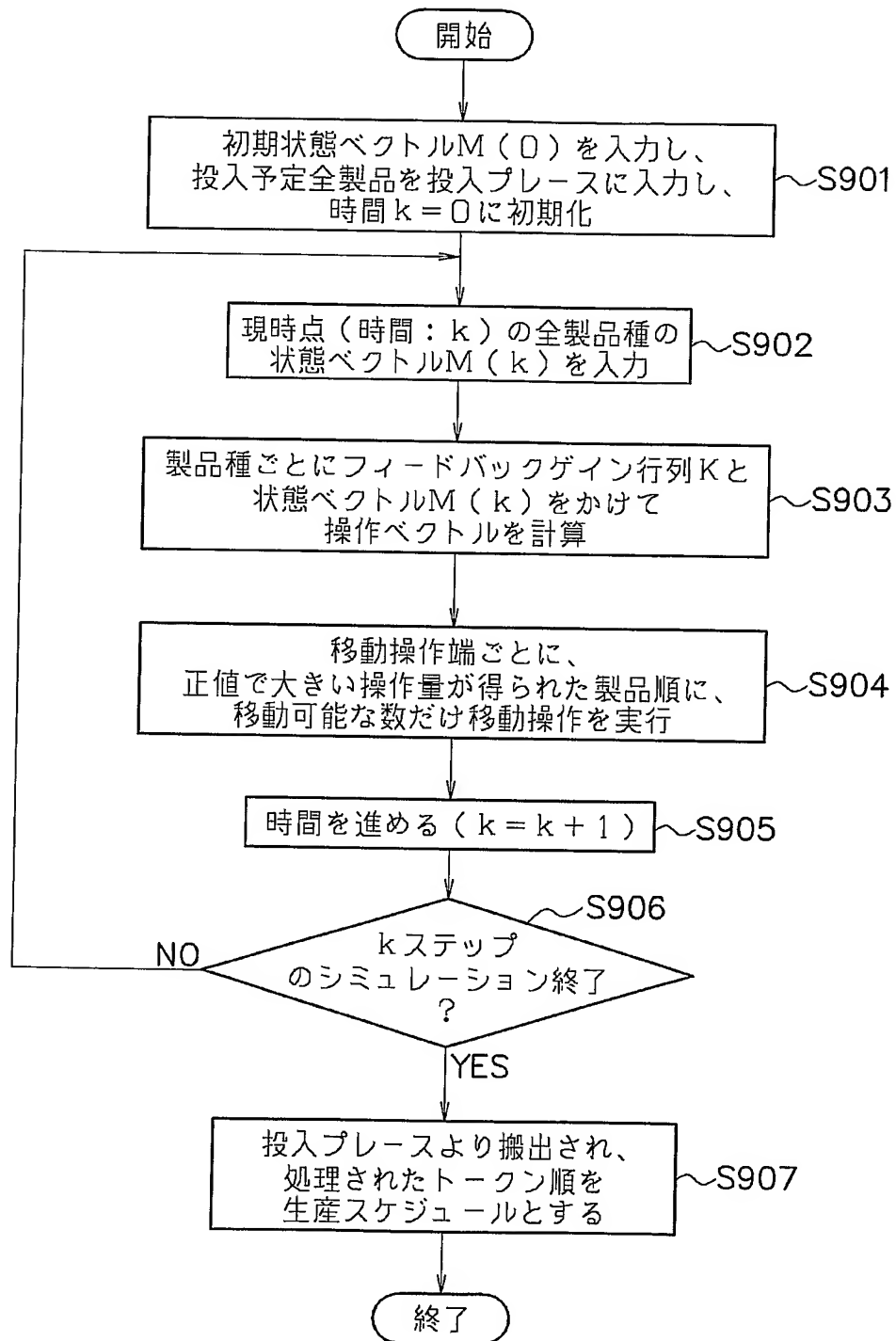


【図 13】

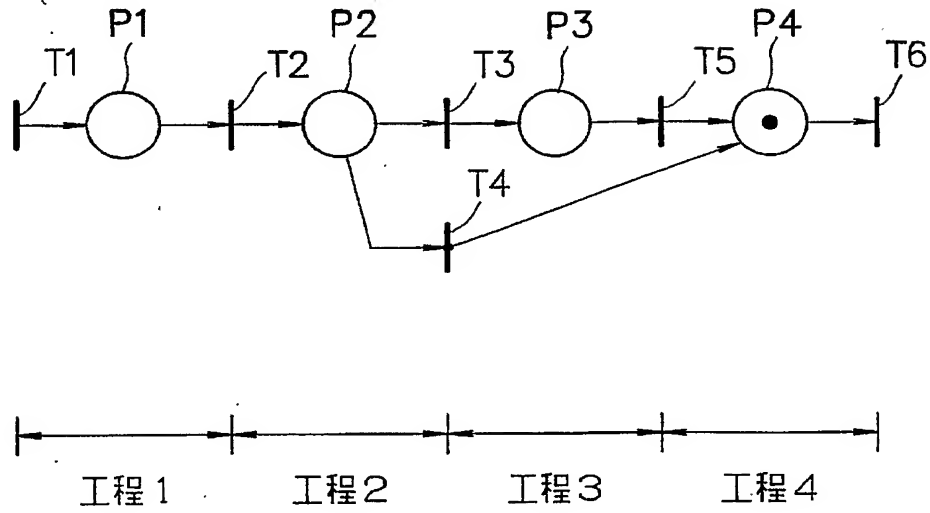




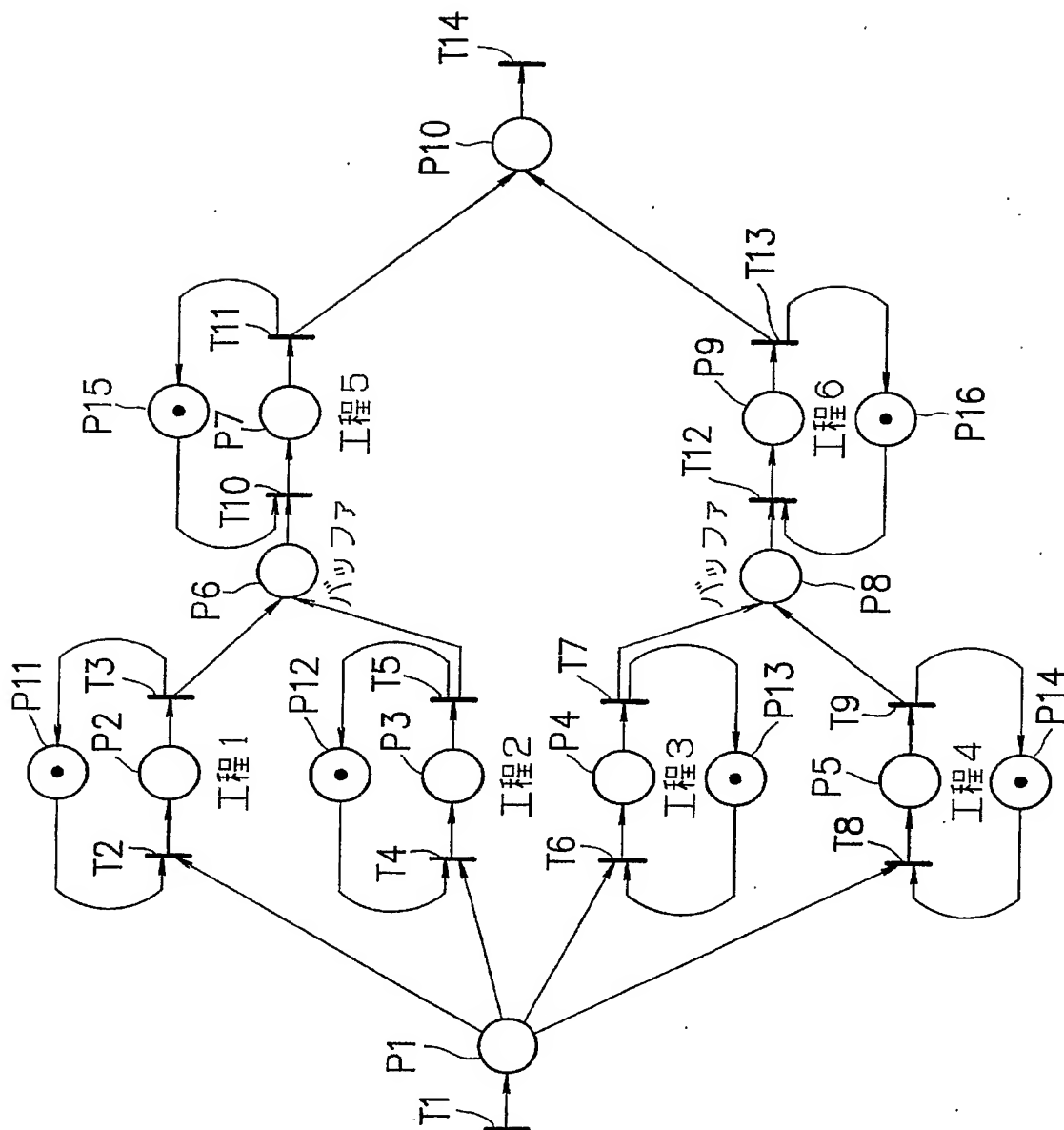
【図 14】



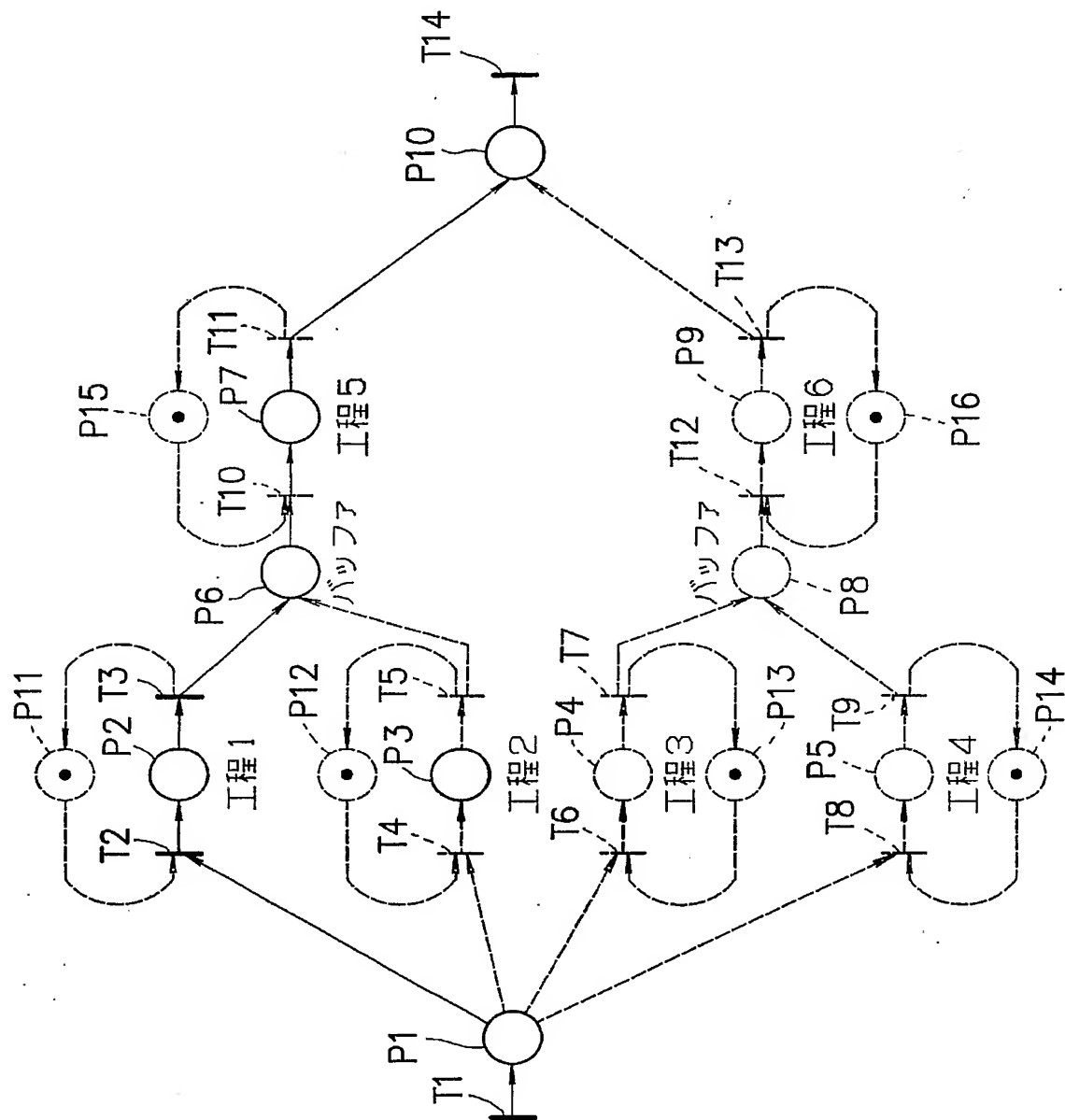
【図 15】



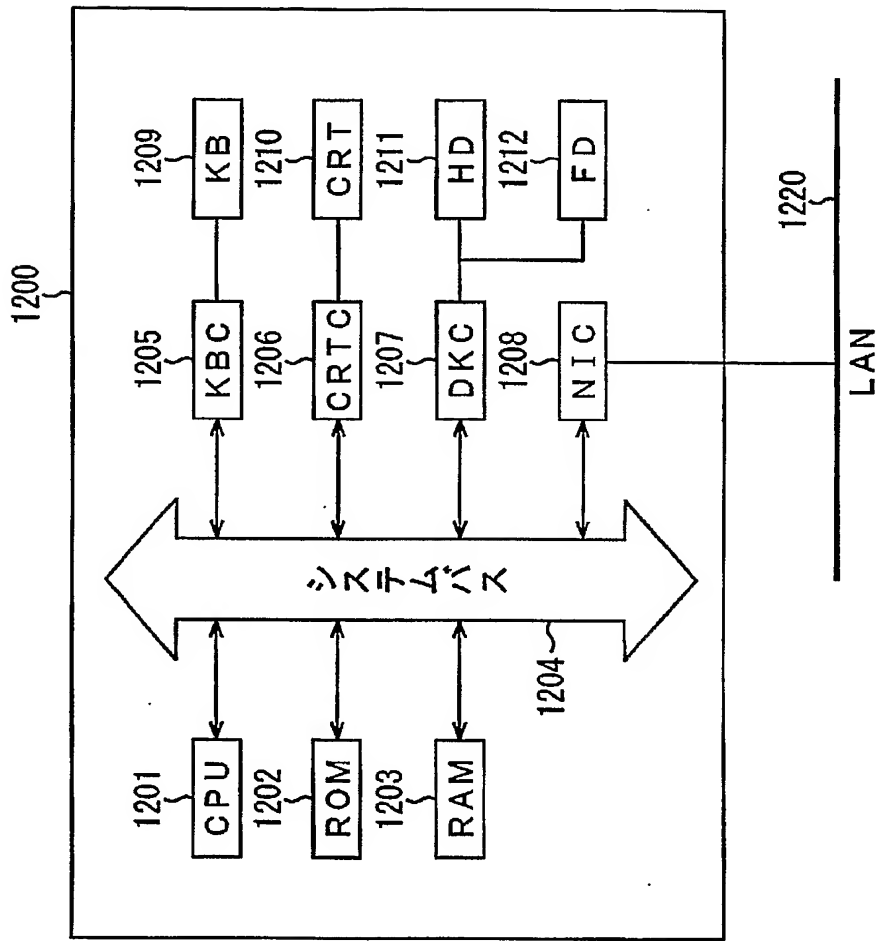
【図 16】



【図 17】



【図 18】



【書類名】 要約書

【要約】

【課題】 実時間で最適な解を得ることが可能な生産・物流スケジュール作成装置を提供する。

【解決手段】 生産・物流プロセスを模擬する離散系の生産・物流シミュレータ 1 a と、生産・物流プロセスの物流状態と物流制約とを数式で表現した状態方程式を用いた数式モデル 3 を保持する数式モデル保持装置と、数式モデル 3 に対して所定の評価関数を用いて最適化計算処理を行って生産・物流シミュレータ 1 a に対する物流指示を算出する最適化計算装置 2 とを有し、最適化計算処理によりフィードバックゲインを算出し、そのフィードバックゲインと物流状態とを用いて物流指示を算出して生産・物流シミュレータ 1 a に与えて、シミュレーションを進め、新たな物流状態を得て、この新たな物流状態をもとに新たな物流指示を算出する処理を繰り返して得たシミュレーション結果から生産・物流プロセスの生産・物流スケジュールを作成する。

【選択図】 図 1

特願 2 0 0 4 - 1 1 4 6 3 7

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 0 0 6 6 5 5 ]

1. 変更年月日

1 9 9 0 年 8 月 1 0 日

[変更理由]

新規登録

住 所

東京都千代田区大手町 2 丁目 6 番 3 号

氏 名

新日本製鐵株式会社